高等学校“十一五”规划教材

Visual Basic 程序设计

主 编 王国权 顾泽元

煤炭工业出版社

责任编辑：高　专　胡玉雁
封面设计 艺铭DESIGN

ISBN 7-5020-2964-8
9 787502 029647 >

ISBN 7-5020-2964-8/TP312
社内编号：5763　定价：32.00元

高等学校“十一五”规划教材

Visual Basic 程序设计

主　编　王国权　顾泽元

副主编　张绍兵　董　军

主　审　石　岩

煤炭工业出版社

·北　京·

内 容 提 要

本书以 Visual Basic 6.0 软件系统为环境，系统地介绍了 Visual Basic 程序设计的基本概念、理论和方法。全书共分为 12 章，内容包括 Visual Basic 程序设计概述、Visual Basic 程序设计基础、Visual Basic 程序设计结构、常用控件、数组及应用、过程、应用程序界面设计、文件操作、绘图应用程序设计、数据库管理应用程序设计、多媒体应用程序设计、网络应用程序设计等。本书既注重理论知识，又突出实践和应用，深入浅出，有利于学生对基础知识的理解和编程能力的提高。

本书是高等学校非计算机专业本、专科学生学习面向对象程序设计的通用教材，也可供参加计算机等级考试的读者学习与参考。

图书在版编目(CIP)数据

Visual Basic 程序设计/王国权，顾泽元主编 .—北京：煤炭工业出版社，2006.11

高等学校“十一五”规划教材

ISBN 7-5020-2964-8

Ⅰ.V… Ⅱ.①王…②顾… Ⅲ.BASIC 语言－程序设计－高等学校－教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2006）第 116453 号

煤炭工业出版社 出版
（北京市朝阳区芍药居 35 号 100029）
网址：www.cciph.com.cn
北京京科印刷有限公司 印刷
新华书店北京发行所 发行

*

开本 787mm×1092mm 1/16 印张 18 3/4
字数 453 千字 印数 1—6,000
2006 年 12 月第 1 版 2006 年 12 月第 1 次印刷
社内编号 5763 定价 32.00 元

前　言

根据教育部高等学校计算机教学指导委员会2003年发布的《关于进一步加强高校计算机基础教学的几点意见》指导精神，按照“1＋X”课程体系改革的要求，我们编写了《Visual Basic程序设计》这本教材。其内容反映了面向对象的程序设计方法，目的是满足高校非计算机专业公共计算机基础课教学的需要。本书教学内容丰富，结构合理，实践性强，深入浅出，既注重基本概念、基本理论、基本方法，又突出实践性和实用性。通过使用本教材，可以使学生掌握Visual Basic程序设计的基本概念、基本理论、基本方法和基本应用技能，培养学生分析问题和解决问题的能力、创新精神和实践能力，为应用计算机解决本专业领域相关问题奠定基础。

本书兼顾实际教学内容的需要和全国计算机等级考试二级Visual Basic考试大纲(2004版)，并结合了高校课程建设与教学改革的新成果。在讲解基本概念、基本理论、基本方法的基础上，通过应用实例或例题进行程序设计实践，操作步骤清晰，有利于学生编程能力和自学能力的提高。本书章节安排合理，符合教学过程和学生学习的实际要求。每章后都配有大量习题，并在附录中给出了全国计算机等级考试二级Visual Basic考试大纲、考试样题、程序调试、错误处理及编程规范，有利于学生的基本训练和复习，使学生掌握程序调试和编程规范等基本技能，培养学生良好的编程规范。

本书按70学时编写，其中实验不少于30学时，各学校可根据教学大纲的要求适当取舍。

本书由王国权、顾泽元任主编，张绍兵、董军任副主编，石岩主审。全书由王国权统稿。其中，第1章、第2章、第4章由王国权编写，第8章、第9章、第10章由顾泽元编写，第3章、第5章、第6章由张绍兵编写，第7章、第11章、第12章由董军编写，附录由石岩编写。在本书编写过程中，得到了黑龙江科技学院各级领导的大力支持和帮助，在此表示感谢！

由于编者水平有限，书中难免有疏漏之处，敬请各位专家和读者批评指正。

编　者

2006年9月

目　录

第 1 章　Visual Basic 程序设计概述

本章主要介绍程序设计涉及的基本概念、面向对象的程序设计方法的基础知识，以及 Visual Basic 6.0 的集成开发环境、窗体的组成、程序管理结构等，并以一个简单的 Visual Basic 6.0 应用程序设计实例说明 Visual Basic 程序设计的基本步骤。

1.1　程序设计的基本概念

在进行程序设计之前，应了解程序设计的基本概念，包括程序、算法、程序设计语言、程序设计等基本知识。

1.程序

程序是操作计算机完成特定任务指令的集合，程序由程序设计语言来实现。Visual Basic 语言就是一种程序设计语言，是用来编写程序的。为了满足人们各种特定的需要，计算机工作者开发了各种工具软件，例如文字处理、表格处理、图形处理、多媒体管理以及各种系统管理工具等，都是一些专用的程序集，用户在界面上与计算机进行交流，就是在调用程序集中的子程序。然而，人们要求计算机做的工作千差万别，现代社会对计算机的要求也是无止境的。计算机厂家不断地制造功能越来越强大的计算机系统，软件开发者也不断地设计出满足不同需要的应用程序。不管计算机结构如何，它要干什么工作，人们总是用某种形式的程序与计算机进行交流的。

2.算法

算法是对解决问题步骤的描述，或者说用来描述程序的实现步骤。算法本身也可以采用不同的方式进行描述，常用描述算法的方法有自然语言描述、伪代码描述、程序流程图描述等，其中程序流程图分为美国国家标准学会(ANSI)提出的标准流程图和两位美国学者提出的 N－S结构流程图。一般情况下，本书采用标准流程图来描述程序的算法。

3.程序设计语言

程序设计语言是由字、词和语法规则构成的指令系统，也称为算法语言。它是人与计算机交流的工具。用户使用程序设计语言，可以告诉计算机做什么，计算机就会根据指令的要求逐条地执行，并且把执行结果告诉用户。

计算机应用的领域很广泛，为了适应不同的需要，往往程序设计语言又各具特点，如有适合于编写系统软件的语言、进行科学计算的语言、数据库管理的语言、图形设计的语言以及人工智能的语言等，还有一些语言具备多种功能。从应用角度说，难以对程序设计语言进行严格分类。随着计算机科学的发展及应用领域的迅速扩展，各种语言的版本都在不断地变化，功能在不断更新和增强，每个时期都有一批语言在流行，又有一批语言在消亡。因此，应该掌握程序设计语言中本质性的、规律性的东西。

根据程序设计语言提供的指令能否被计算机直接执行的情况，一般可以把程序设计语言分为如下三种：

(1) 机器语言。机器语言也称二进制语言，它的代码全部由二进制符号“0”和“1”按不同的方式排列组合而组成。用二进制语言编写的程序能够被计算机直接识别执行，是最早期的计算机程序设计语言，与计算机硬件有关。机器语言的特点是执行快，但是二进制代码记忆难度大，易于出错。

(2) 汇编语言。汇编语言也称符号语言，是用一些英文缩写等助记符来表示二进制代码的程序设计语言。用汇编语言编写的程序，只需要经过简单的翻译就可以被计算机执行。汇编语言相对于机器语言而言，容易记忆，增强了程序的可读性，但是不能直接被计算机执行，必须通过一个汇编程序被翻译成机器语言后才能执行。

机器语言和汇编语言都是面向计算机的程序设计语言，每种计算机的 CPU(中央处理器)都有一套自己的指令系统，称为机器语言。机器语言编写的程序执行速度快，资源要求低，通常用于编写直接与机器交互的程序，如控制程序等。

(3) 高级语言。由人们易于接受的、接近人类语言的描述方式构成的指令系统，称为高级语言。高级语言是以汇编语言为基础逐渐形成的，更加接近人类的自然语言的一种程序设计语言。一种高级程序设计语言往往只有一百几十条词汇、若干条规则，代码简短、便于记忆、易学易用。高级语言提供常用的数据描述和对数据操作的规则描述，这些规则是脱机的，程序员只需要专注于问题的求解，不必关心机器内部结构和实现。

通常所说的“程序设计语言”是指高级语言，用高级语言编写的程序称为“源程序”。计算机不能直接识别源程序代码，必须翻译成二进制程序代码才能在机器上运行。翻译方式有两种：一种称为解释方式，另一种称为编译方式。解释方式是由解释程序对源程序逐个语句一边翻译，一边执行，这种方式执行速度慢，便于观察调试程序；编译方式是由编译程序把源程序全部翻译成二进制程序，编译后的程序称为“目标程序”，一旦编译成功，目标程序就可以反复高速执行。每种高级语言都配有解释或编译系统，解释方式交互性强，而编译方式速度更快。Visual Basic 提供解释和编译两种执行方式。源程序和目标程序都可作为文件保存下来。

Visual Basic 语言是 20 世纪 90 年代 Microsoft 公司以结构化 Basic 语言为基础，以事件驱动为运行机制而开发的一种通用的可视化程序设计语言。除了提供常规的编程环境外，还提供了可视化设计工具，便于程序员建立图形对象，巧妙地把 Windows 编程的复杂性“封装”起来。它不仅具有传统的程序设计语言的功能，而且随着版本的改进，功能越来越强大，不但可以作为多媒体软件制作工具、实现数据库管理，而且还具有网络功能等。

4.程序设计

程序设计是指为了利用计算机解决某种特定的问题，使用某种程序设计语言，编写计算机执行的指令序列。程序设计是一项在遵守程序设计语言语法前提下的创造性工作，一般需要完成数据描述和数据处理等工作。

(1) 数据描述。数据描述是指把被处理的信息描述成计算机可以接受的数据形式，如整数、实数、字符、数组等。使用计算机进行信息处理时，必须把信息转换成可被机器识别的“数据”，如数字、文字、图形、声音等。不管什么数据，计算机都以二进制形式进行存储和加工处理。数据是信息的载体，信息依靠数据来表达。有些数据，可以通过程序设计语言直接用“数据类型”描述，如数值和字符等；有些数据，一般的程序设计语言没有提供直接定义，但可以作

为外部文件使用，如 Visual Basic 可以在界面或程序代码中使用图形、声音等文件。

(2) 数据处理。数据处理是指为了获取所需的数据和有用的资料而对数据进行输入、输出、整理、计算、存储、维护等一系列处理工作。通常可用一个子程序来实现数据的某种操作，数据可以通过外部设备，如键盘、磁盘输入，也可以在程序内部使用初始化、赋值等方式获得；程序的执行结果可以输出到屏幕、打印机、文件，或者传送给其他程序。

在程序设计中，编写源程序一般常出现三类错误：一是编译错误，即编译程序时发现的语法错误，如表达式不符合语法规则等；二是运行错误，即执行目标程序时发现错误，如除数为0等；三是逻辑错误，即编译和运行时均不能发生的错误，如由表达式书写错误带来程序运行结果错误等。因此，程序一般要经过反复的调试、验证才能完善投入使用，编写的程序应力求具备正确性、易读性、运行高效性。

1.2 面向对象的程序设计方法

目前程序设计方法主要分为两类：一类是结构化的程序设计方法，另一类是面向对象的程序设计方法。

结构化的程序设计方法是1966年被提出的，其思想是把程序设计分为顺序结构、选择结构、循环结构。每种结构的程序流程不同，当需要按照语句的先后次序执行时，采用顺序结构；当需要根据条件选择地执行程序时，采用选择结构；当需要根据条件成立与否来决定是否循环执行程序时，采用循环结构。不论程序简单，还是复杂，都可以通过上述3种基本结构来实现，提高了程序的清晰度和可维护性。程序设计结构的内容详见第3章。结构化的程序设计方法是面向数据和程序的方法，这种方法把数据和程序作为相互独立的实体，在编写程序时，使数据和程序保持一致是程序员的一个沉重的负担。

面向对象的程序设计方法是在结构化的程序设计方法基础上发展起来的，与结构化的程序设计方法有本质的不同，它是把数据和程序组合起来作为一个对象，每个对象除了传递消息之外，相互之间没有其他联系。因此，使程序员摆脱了具体的数据格式和程序的束缚，可以集中精力研究和设计要处理的对象。

在 Visual Basic 中，体现了面向对象程序设计和结构化程序设计两种思想，即总体是面向对象的程序设计思想，而在每个对象内部编程时则采用结构化的编程思想。

1. 类

抽象是人认识事物时的一种行之有效的方法，通过抽象可以反映出诸多事物共同拥有的本质特征。程序设计语言也可以进行不同程度的抽象，结构化程序设计语言主要依靠基本结构对程序进行抽象，编程人员需要考虑整个程序的流程，依据流程的不同采用不同结构实现程序的功能。在面向对象的程序设计中，抽象的方法是通过“类”的概念来体现的，“类”是具有共同属性和操作的多个对象的相似特征的统一描述，反映了一个群体事物的共同数据特征和功能特征。“类”可以分多个层次抽象，较高抽象层次的类称为“父类”，较低抽象层次的类称为“子类”。例如，“车”可以抽象成一个类，这类事物具有一些共同特性。如果把“车”作为父类，则可以把“轿车”、“赛车”、“摩托车”等作为子类，每个子类中的事物也具有共同特性，不同子类中的事物都具有一部分不同的特性。

2. 对象

在 Visual Basic 中,对象是指程序和数据的组合。简单地说,就是把对象当作一个单位来处理。"类"是抽象的概念,类中的每个具体事物称为该类的"对象"。对象是类的实例,所以每个对象都有自己的特性值。对象是动作的主体,一个复杂的对象可以由若干个简单的对象组成。在 Visual Basic 中,对象主要包括"窗体"和"控件"等。窗体是程序运行时的一个窗口;控件是窗体上的界面元素,如命令按钮和文本框等。每个对象的静态特征有对象的"属性",动态特征有对象的"事件"和"方法"。在设计对象时,只要将对象添加在界面上,不必编写建立和描述对象的程序代码。

3. 对象的属性

对象的属性是对象的数据,用来描述对象的静态特征。属性有属性值,改变对象的属性值就可以改变对象的特性,如窗体对象的 Width 和 Height 属性用来决定其宽度和高度。不同的属性取值决定了这个对象不同于其他对象。每个对象包含的属性不同,但有些属性是很多对象共同有的,如 Width 和 Height 属性。

在 Visual Basic 程序中,设置对象的属性,要按照如下的引用格式:

对象名.属性名 = 属性值

例如,一个窗体对象的名字是 Form1,则通过程序代码设置其高度的语句是:

Form1.Height = 1000

说明:每个对象都有一个惟一的标识,也就是对象的"名称"属性。要引用哪个对象,就要引用这个对象的名称,如果省略对象名称则表示对当前窗体属性进行设置。

4. 对象的方法

对象的方法是指对象可以进行的动作或行为,用来描述对象的动态特征。对象的方法是系统已经实现的一些函数,用户只需要直接使用即可。引用对象的方法的语法格式如下:

对象名.方法名[参数列表]

例如,窗体对象有 Print 方法,可以在窗体上显示数据,在窗体 Form1 上打印输出"计算机程序设计"的语句是:

Form1.Print "计算机程序设计"

说明:如果省略对象名则表示引用当前窗体的方法。

5. 对象的事件

在 Visual Basic 中,每个对象都有一些系统预先定义好的、对象可以执行的动作,这些能被对象识别的动作称为事件,事件用来描述对象的动态特征。事件能够触发对象进入活动状态,当对象被事件触发时就可以执行对应的事件程序代码。

Visual Basic 为事件驱动的编程机制,应用程序是由事件驱动的,即只有当事件发生时,响应事件的程序代码才会运行。Visual Basic 程序运行的过程就是通过事件触发某个对象,随着该对象的活动又触发新的事件,新事件又触发另一个对象,对象就是以这种方式联系在一起的,如果没有事件发生,则整个程序就处于停滞状态。Visual Basic 编程的核心就是为每个事件编写响应事件的过程代码,不同对象响应不同事件的过程就构成了应用程序。

事件也有自己的名称,如命令按钮具有"单击事件",事件名称的关键字是 Click。当用户单击命令按钮时,就会触发 Click 事件。其实每个事件都是一个函数,当事件被触发时,Visual Basic 就会转去执行该函数中的代码,实现相应的功能。如果在程序运行时单击命令按钮来完

成相应的功能,就要在该命令按钮的 Click 事件中编写代码。在 Visual Basic 中,事件的语法格式如下:

```
Private Sub 对象名_事件名()
  事件过程代码
End Sub
```

例如,命令按钮对象的名称是 Command1,其 Click 事件的代码如下:

```
Private Sub Command1_Click()
  事件过程代码
End Sub
```

注意:窗体的事件格式为 Form_事件名()。

1.3 Visual Basic 6.0 简介

Visual Basic(Basic, Beginner's All-purpose Symbolic Instruction Code)是由 Microsoft 公司开发的可视化、面向对象的程序设计语言。由于 Visual Basic 具有简单易学、可视化、面向对象、事件驱动的特性而受到用户欢迎,也成为专业人员开发 Windows 应用程序得心应手的开发工具。

1.3.1 Visual Basic 6.0 的功能特点及版本

1. Visual Basic 6.0 的功能特点

Visual Basic 具有强大的 Windows 应用程序开发功能,具有以下主要特点:

(1) 面向对象的可视化编程工具。是指应用面向对象的程序设计方法,把程序和数据封装成一个对象,并且大多数对象都是可视化的。用户可以利用这些对象方便地设计应用程序的图形界面,而不需要编写大量的程序代码去描述界面元素的外观和位置,并且可以通过对象属性的设置及方法和事件的使用,灵活地设计程序。

(2) 采用事件驱动编程机制。开发人员通过编写对象事件过程的程序代码来响应用户对对象的操作,在响应不同的事件时执行不同的代码。事件可以由用户操作触发,也可以由来自操作系统或其他应用程序的消息触发,甚至可以由应用程序本身的消息触发。事件的顺序决定了程序代码的执行顺序,应用程序每次运行时执行程序代码的顺序是不同的。每个事件都能驱动一段程序的运行,程序员的主要任务就是编写响应每个事件的程序代码。

(3) 集成的开发环境。Visual Basic 提供了集成开发应用程序的环境,用户可以在这个环境中,完成应用程序的界面设计、编写程序代码、调试程序、发布程序等所有程序设计步骤。

(4) 强大的数据库管理功能。Visual Basic 利用数据库控件可以直接或间接地访问数据库管理系统,如 Microsoft Access、Microsoft SQL Server、Oracle 等,Visual Basic 6.0 中增加了强大 ADO(ActiveX Database Object)控件,其使用更加方便,访问速度更快。

(5) ActiveX 技术。Visual Basic 支持对象链接与嵌入(OLE)技术,借助这种技术可以更方便地开发 Visual Basic 应用程序和扩展应用程序的功能。

(6) 联机帮助功能。与其他 Windows 软件一样,在 Visual Basic 中,可以利用菜单或在程序设计时按 F1 键,方便地获得 MSDN(Microsoft Develop Network Library)的帮助信息。

2. Visual Basic 6.0 的版本

Visual Basic 是随着 Windows 3.0 操作系统的推出而产生的。1991 年推出 Visual Basic

1.0 版,1992 年 11 月发布了 Visual Basic 2.0 版,1993 年 5 月推出了 Visual Basic 3.0 版,1995 年 9 月推出了 Visual Basic 4.0 版,1997 年 3 月发布了 Visual Basic 5.0 版,1998 年发布了 Visual Basic 6.0 版。目前 Visual Basic 6.0 版包含 3 个版本,可供不同需要的用户使用。

(1) 学习版(Learning Edition):是 Visual Basic 6.0 的基础版本,功能最简单,可以使学习编程的人员轻松入门和开发 Windows 应用程序。

(2) 专业版(Professional Edition):在学习版的基础上,增加了 ActiveX 控件、Internet 控件等工具,为专业人员提供了功能强大的开发环境。

(3) 企业版(Enterprise Edition):在专业版的基础上,增加了开发分布式应用程序等功能,并包括整套的 BackOffice 工具。

本书是以 Visual Basic 6.0 中文企业版为开发环境进行介绍的。

1.3.2 Visual Basic 6.0 的安装、启动及退出

1.Visual Basic 6.0 安装的硬件环境要求

(1) 处理器:80486 或者更高的处理器。

(2) 内存:对于 Windows NT 或更高版本的操作系统,至少需要 32 MB 内存。

(3) 硬盘:学习版的典型安装需要 48 MB 空间,完全安装需要 80 MB 空间;专业版的典型安装需要 48 MB 空间,完全安装需要 80 MB 空间;企业版的典型安装需要 128 MB 空间,完全安装需要 147 MB 空间;另外安装 MSDN 帮助文档需要 67 MB 空间。

(4) 显示器:VGA 或者更高分辨率的显示器。

(5) 光驱:8 倍速以上的 CD-ROM。

2.Visual Basic 6.0 安装的软件环境要求

(1) 操作系统:Microsoft Windows 95/98,Microsoft Windows NT4.0(Pack3)、Microsoft Windows XP 或者更新版本的操作系统。

(2) 浏览器:Microsoft Internet Explore 4.01 或者更高的版本。

2.Visual Basic 6.0 的安装过程

硬件环境和软件环境都符合安装条件后,一般安装过程需要经过以下步骤:

(1) 将 Visual Basic 6.0 中文企业版光盘放入光驱,启动后显示"Visual Basic 6.0 中文企业版安装向导"对话框,如图 1-1 所示。

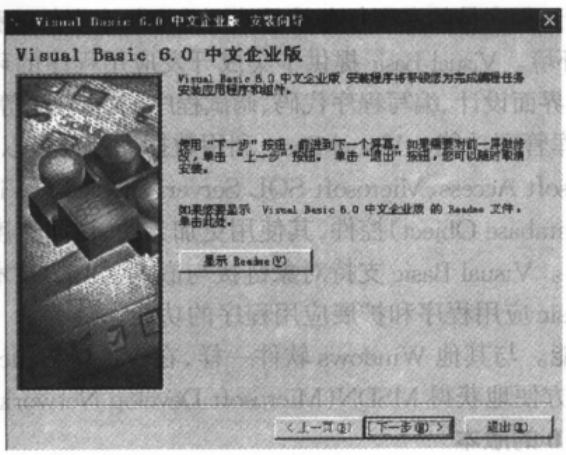


图 1-1 "Visual Basic 6.0 中文企业版安装向导"对话框

(2) 在图 1－1“Visual Basic 6.0 中文企业版安装向导”对话框中，单击“下一步”按钮，打开“最终用户许可协议”对话框，如图 1－2 所示。

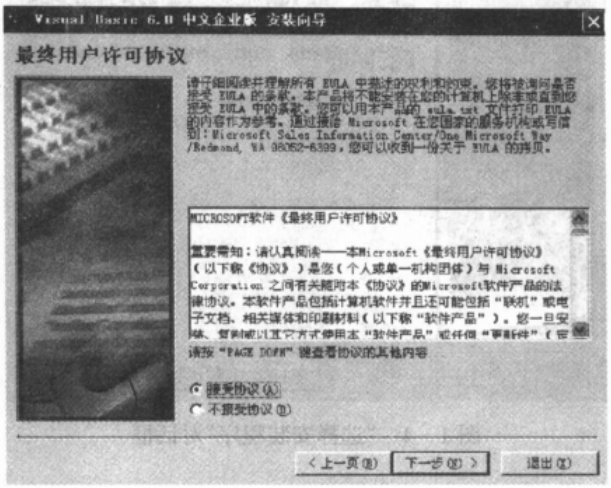


图 1－2　“最终用户许可协议”对话框

(3) 在图 1－2 的“最终用户许可协议”对话框中，选择“接受协议”后，单击“下一步”按钮，打开“产品号和用户 ID”对话框，如图 1－3 所示。

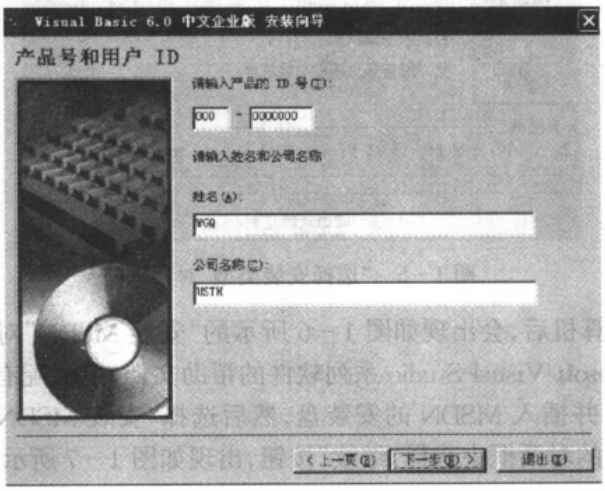


图 1－3　“产品号和用户 ID”对话框

(4) 在图 1－3 的“产品号和用户 ID”对话框中，输入产品 ID 号和用户信息后，单击“下一步”按钮，打开如图 1－4 所示的“选择安装程序”对话框。

(5) 在图 1－4 的“选择安装程序”对话框中，选择“安装 Visual Basic 6.0 中文企业版”，然后单击“下一步”按钮，出现如图 1－5 所示的对话框。如果选择“典型安装”系统会自动安装一些最常用的组件；如果选择“自定义安装”，用户可以根据实际需要有选择地安装组件。然后安装程序将文件复制到硬盘中，完成 Visual Basic 6.0 中文企业版的安装，并要求重新启动计算机，更新系统配置以使安装程序生效。

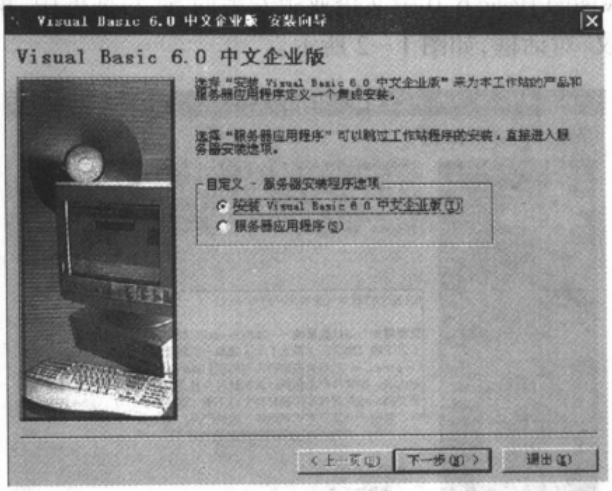


图 1-4 “选择安装程序”对话框

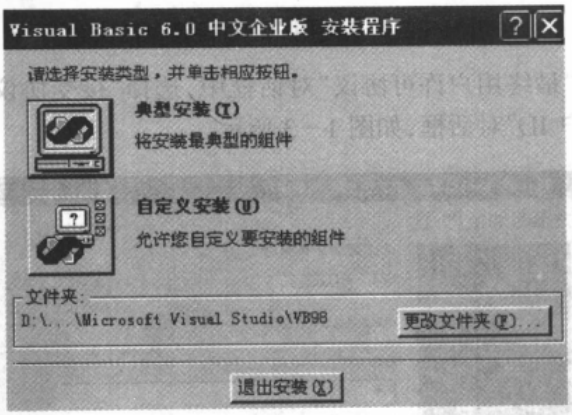


图 1-5 “选择安装类型”对话框

(6) 重新启动计算机后，会出现如图 1-6 所示的“安装 MSDN”对话框，要求安装 MSDN。MSDN 是 Microsoft Visual Studio 系列软件的帮助文档，单独配有安装盘。当出现“安装 MSDN”对话框时，并插入 MSDN 的安装盘，然后选择“安装 MSDN”复选框，单击“下一步”按钮，然后在出现的对话框中选择“继续”按钮，出现如图 1-7 所示的“选择安装类型”对话框。

(7) 在图 1-7 的“选择安装类型”对话框中，选择“自定义安装”，出现如图 1-8 所示的“自定义安装”对话框，在对话框中选择需要安装的组件，单击“继续”按钮，完成安装过程。

4. Visual Basic 6.0 的启动

安装成功以后，按照以下步骤可以启动 Visual Basic 6.0 中文企业版，进入其集成开发环境(IDE)中。

(1) 在 Windows 操作系统中，单击“开始”→“程序”→“Microsoft Visual Basci 6.0 中文版”→“Visual Basic 6.0 中文版”，出现如图 1-9 所示的“新建工程”对话框；然后在“新建”选项卡中，选择新建一个“标准 EXE”工程，就可以打开 Visual Basic 6.0 的集成开发环境窗口，如图 1-10 所示。

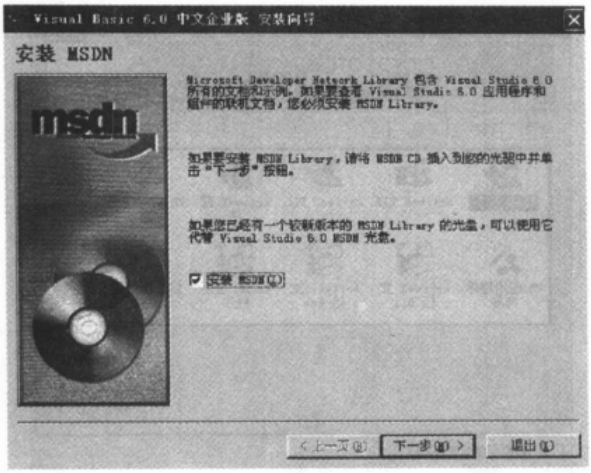


图 1-6　"安装 MSDN"对话框

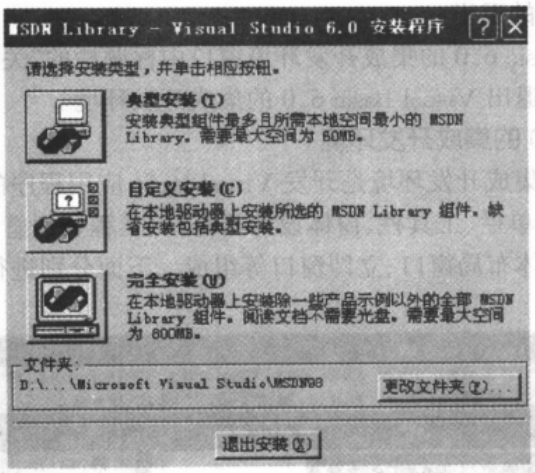


图 1-7　"选择安装类型"对话框

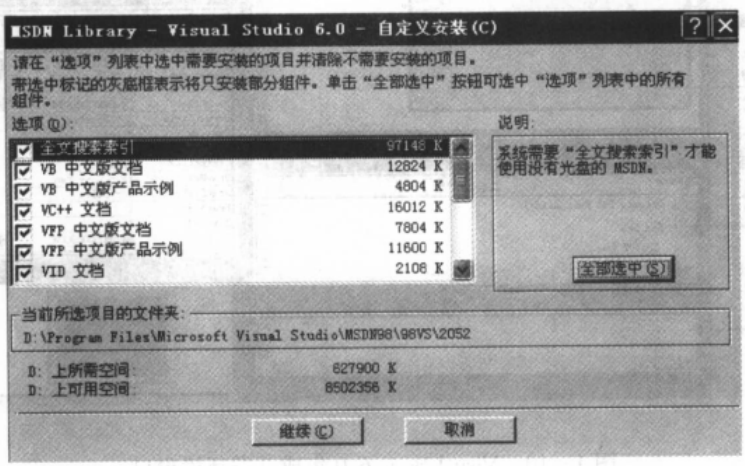


图 1-8　"自定义安装"对话框

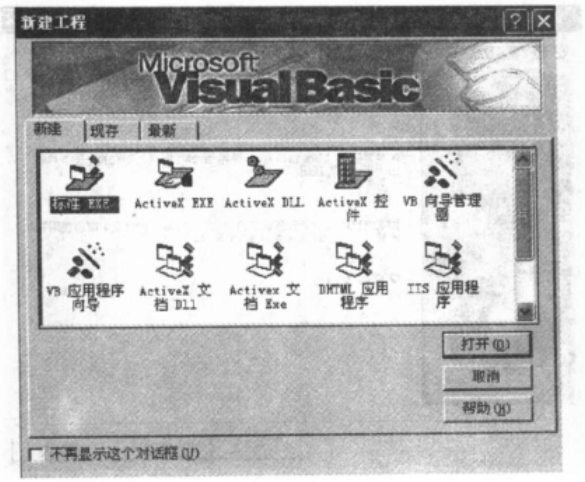


图 1-9 “新建工程”对话框

5. Visual Basic 6.0 的退出

通过单击 Visual Basic 6.0 的集成开发环境窗口中标题栏的“关闭”按钮或者单击“文件→退出”的方法，都可以退出 Visual Basic 6.0 的集成开发环境。

1.3.3 Visual Basic 6.0 的集成开发环境

Visual Basic 6.0 的集成开发环境是开发 Visual Basic 应用程序的平台，其窗口如图 1-10 所示，包括标题栏、菜单栏、工具栏、窗体设计器窗口、工程资源管理器窗口、属性窗口、代码编辑窗口、工具箱、窗体布局窗口、立即窗口等组成。下面分别进行介绍。

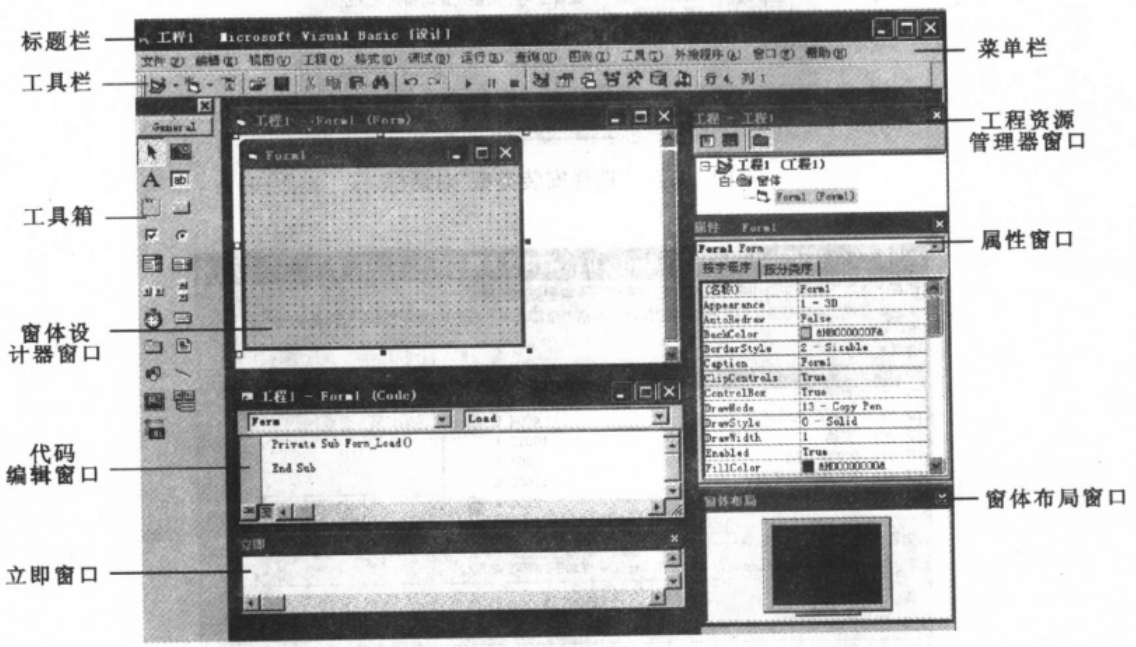


图 1-10 Visual Basic 6.0 集成开发环境窗口

1. 标题栏

在标题栏中,显示2项内容:一是工程的名称,二是Visual Basic集成开发环境的3种工作状态之一。如图1-10所示,启动Visual Basic 6.0后,在标题栏中显示"工程1-Microsoft Visual Basic[设计]"。"工程1"为默认的工程名称,标题栏中显示的[设计]代表当前为设计状态,可以进行用户界面设计、程序代码编写、属性设置等设计工作;标题栏中显示的[运行]代表当前为程序运行状态,看到的是程序运行的结果,此时不可进行界面设计、编写程序代码、属性设置工作;标题栏中显示的[中断]代表当前为中断状态,即程序运行暂时中断,可以查看程序运行的中间结果,此时可以编辑程序代码,但不可编辑程序界面。

2. 菜单栏

菜单栏中包括"文件"、"编辑"、"视图"、"工程"、"格式"、"调试"、"运行"、"查询"、"图表"、"工具"、"外接程序"、"窗口"、"帮助"等菜单,提供了Visual Basic 6.0的所有命令。

3. 工具栏

工具栏为常用菜单命令提供了快捷方式,利用工具栏按钮可以迅速地访问常用的菜单命令。Visual Basic 6.0中有4种工具栏,即"标准"、"编辑"、"窗体编辑器"、"调试"工具栏,其中最为常用的是"标准"工具栏。一般情况下只显示"标准"工具栏,要显示或隐藏其他工具栏,可以通过"视图→工具栏"进行显示或隐藏。

4. 窗体设计器窗口

窗体是一个窗口,用来设计应用程序的界面。是各种控件的容器,用户可以通过在窗体中添加各种控件,进行应用程序的界面设计。

5. 工程资源管理器窗口

工程资源管理器窗口类似于Windows的资源管理器,管理应用程序包含的所有工程及其文件。工程资源管理器窗口中包含"查看代码"、"查看对象"、"切换到文件夹"3个按钮,可以通过进行切换操作,分别进入代码编辑窗口、窗体设计窗口、显示文件夹状态。

6. 属性窗口

属性窗口用于设置对象的属性值,在属性窗口中,列出了当前选定对象的属性名和属性值,用户可以根据设计要求进行属性值的设定和修改。属性窗口由标题栏、对象下拉列表框、选项卡、属性列表、属性说明等组成。

7. 代码编辑窗口

代码编辑窗口是用来显示、编写和修改程序代码的。当双击窗体或控件时,就进入了代码窗口,或者在工程资源管理器窗口中,单击"查看代码"按钮也可以进入代码编窗口。

8. 工具箱

工具箱中包含若干个图标,每个图标都是Visual Basic应用程序的控件。新建一个"标准EXE"工程后,工具箱里只包含Visual Basic 6.0内部的标准控件,用户可以在工具箱箱中添加其他控件。

9. 窗体布局窗口

窗体布局窗口用于调整窗体在屏幕上显示的位置,在该窗口中,可以通过拖动表示当前窗体的图标,调整程序运行时窗体的位置。

10. 立即窗口

立即窗口用于调试程序,可以在程序中断状态查看程序的中间结果。在立即窗口中输

入一条语句后，按回车键就可以立即执行该条代码。

1.4 Visual Basic 6.0 应用程序设计步骤

在熟悉 Visual Basic 6.0 集成开发环境的基础上，要掌握 Visual Basic 6.0 窗体的组成、属性、方法及事件，程序管理，以及程序设计的一般步骤，才能进一步学习 Visual Basic 应用程序的设计。

1.4.1 Visual Basic 6.0 的窗体

窗体(Form)是 Visual Basic 的对象，是控件的容器，应用程序界面的设计是从窗体开始的。在 Visual Basic 工程中，每个窗体是工程中的一个模块，它单独保存在一个扩展名为 frm 的文件中，其文件内容包括该窗体及控件的属性信息、程序代码等。

1. 窗体的组成

如图 1-11 所示，窗体由标题栏、工作区、“最小化”按钮、“最大化”按钮、“关闭”按钮、菜单组成。

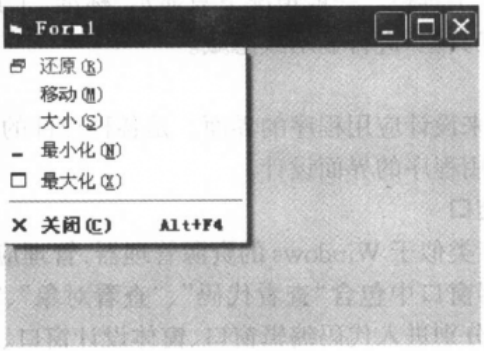


图 1-11 窗体的组成

2. 窗体的属性

窗体具有多种属性，这些属性值的设置结果决定了窗体的特征。窗体常用的属性如下：

(1) Name 属性：设置窗体的名称，是在程序代码中的惟一标识，它不能为空，在程序运行时不能修改，一般窗体的命名应体现其功能。

(2) Caption 属性：设置窗体标题栏中的文本。

(3) Enable 属性：设置窗体是否可用。取值为 True 时可用；取值为 False 时不可用，表现为窗体的标题栏为灰色显示。

(4) BorderStyle 属性：设置窗体边框的样式。取值为 0、1、2、3、4、5，默认值为 2。

(5) Visible 属性：设置窗体在程序运行时是否可见。

(6) AutoRedraw 属性：设置是否重画窗体上显示的图形和文字。

(7) BackColor 属性：设置窗体工作区的背景颜色。

(8) ForeColor 属性：设置窗体工作区的前景颜色。

(9) Font 属性：设置窗体上输出字符的字体、字形、字号及窗体上控件的 Font 属性的默认值。

(10) Top属性:设置窗体上边界与屏幕上边界之间的相对距离。

(11) Left属性:设置窗体左边界与屏幕左边界之间的相对距离。

(12) Width属性:设置窗体的宽度。

(13) Height属性:设置窗体的高度。

3. 窗体的方法

(1) Print方法:用于在窗体、图片框(PictureBox)、打印机(Printer)、立即窗口(Debug)中输出数据。其语法格式如下:

[对象名称.]Print[表达式列表][,|;]

例如:

```
Print "计算机程序设计"                    '在当前窗体上输出计算机程序设计
Picture1.Print "计算机程序设计"            '在图片框中输出计算机程序设计
Printer.Print "计算机程序设计"             '在打印机中输出计算机程序设计
Debug.Print "计算机程序设计"               '在立即窗口中输出计算机程序设计
```

语法格式中的"表达式列表"为输出的内容,可以是数值表达式或者字符表达式。如果为数值表达式,则输出该表达式的值;如果为字符表达式,则输出字符表达式原样。

一般情况下,执行一次Print语句后就要自动换行。如果要在同一行中输出多个表达式的值或字符串,可以在语句末尾加上一个分号或一个逗号。分号表示"紧凑格式"输出,即连续输出;逗号表示"松散格式"输出,即输出位置分别分布在当前行中连续的输出区中。

(2) Cls方法:用于清除在窗体上输出的数据或者图形,也可以用于清除在图片框(PictureBox)等对象中输出的数据。

(3) Show方法:用于显示一个窗体。可以带有参数0或1。0为无模式显示方式,1为模式显示方式。

(4) PrintForm方法:用于从打印机上输出在窗体中当前显示的内容。

(5) Hide属性:用于隐藏窗体。

(6) Move属性:用于移动窗体,同时可以改变窗体的大小。

(7) Refresh属性:用于刷新窗体中显示的内容。

4. 窗体的事件

(1) 加载(Load)事件:窗体加载时触发。

(2) 单击(Click)事件:单击窗体的空白区域时触发。

(3) 双击(DblClick)事件:双击窗体的空白区域时触发。

(4) KeyPress、KeyDown、KeyUp事件:当按键盘上的键时触发。

(5) Resize事件:当窗体大小发生变化时触发。

(6) Unload事件:卸载窗体时触发。

1.4.2 Visual Basic 6.0的程序管理结构

在高级语言程序设计中,每种高级语言都有自己的结构,以便对程序的各个组成部分进行有效的管理。应用程序的管理结构是指组织程序代码的方法,它决定着程序代码存放的位置和执行的顺序。Visual Basic是以工程为单位创建的,并通过"工程资源管理器"、"工程属性"窗口实现程序工程的有效管理。

一个Visual Basic 6.0应用程序可以包含多个工程,每一个工程可以包含多种模块,包

括窗体模块、标准模块、类模块等。每个工程、模块都分别对应一个文件，文件的扩展名可以区分它们的种类。Visual Baisc 6.0 应用程序的管理结构如图 1－12 所示。一个 Visual Basic 6.0 应用程序中可能包含如下的文件：

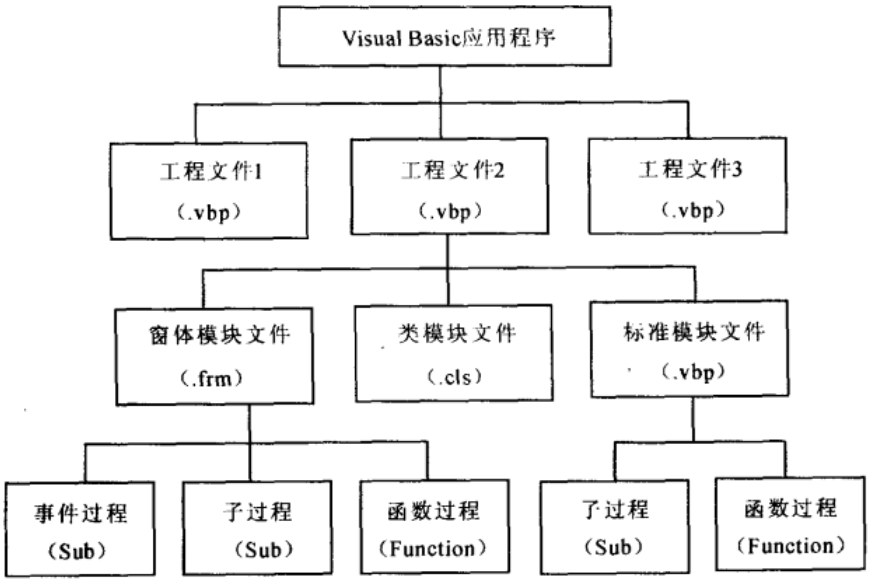


图 1－12　Visual Basic 应用程序管理结构

(1) 工程文件(.vbp)：一个工程对应一个工程文件，一个应用程序中可以包含多个工程文件。

(2) 工程组文件(.vbg)：一个较复杂的应用程序，可能包含多个工程文件，这些工程文件就组成一个工程文件组，称工程组文件。

(3) 窗体文件(.frm)：一个窗体对应一个窗体文件，在窗体文件中，包括窗体及窗体中的控件、窗体及窗体中控件的属性、程序代码等。

(4) 标准模块文件(.bas)：用于定义程序中的通用过程和全局变量，这些通用过程和窗体可以被程序中的所有窗体使用。

(5) 类模块文件(.cls)：用户可以在类模块中定义自己的类。

(6) 资源文件(.res)：用于存放程序所需要的文本、声音、图片等各种资源。

在 Visual Basic 应用程序管理结构中，最为常用的是工程文件和窗体文件。在窗体模块文件、标准模块文件中，包含各种过程，可分为事件过程、子过程、函数过程。

1.4.3　Visual Basic 6.0 的程序管理操作

工程资源管理器窗口是 Visual Basic 6.0 提供的一个可视化程序管理工具，在工程资源管理器窗口中，可以很方便地查看程序代码和对象，切换程序文件夹，同时可以对管理结构中的文件进行添加、删除、切换等操作。

1. 添加窗体

一个工程文件中，可能包含多个窗体文件。当用户需要添加窗体文件时，可以通过 Visual Basic 6.0 的标准工具栏中的“添加窗体”按钮，或者通过如图 1－13 所示的工程资源管理器窗口中的快捷菜单来添加窗体。

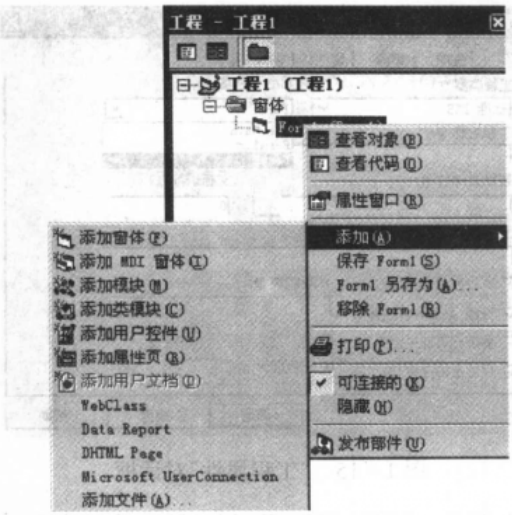


图 1－13　添加、删除窗体快捷菜单

如图 1－14 所示，“添加窗体”对话框中，单击“新建”选项卡，选择“窗体”后，单击“打开”按钮，就可以添加窗体。

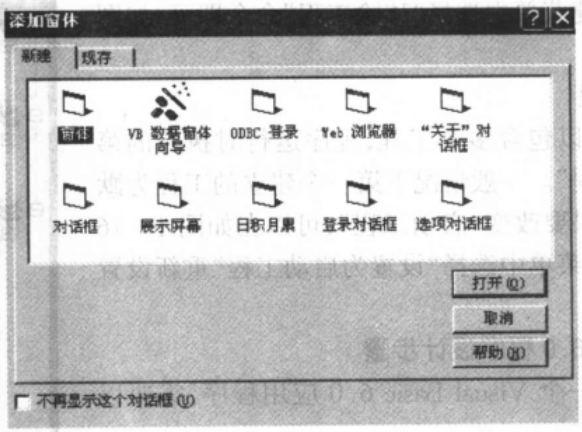


图 1－14　“添加窗体”对话框

2．删除窗体

在工程资源管理器窗口中，在右击需要删除的窗体文件的快捷菜单中，选择“移除窗体 Form”命令，就可以删除该窗体。

3．设置启动窗体

如果一个启动工程中包含多个窗体文件，在默认的情况下，第一个建立的窗体为默认的启动窗体，即从第一个建立的窗体开始运行。如果改变启动窗体，在工程资源管理器窗口中右击工程弹出快捷菜单，在该菜单中选择“工程 1 属性”，出现“工程 1 属性”对话框，如图 1－15 所示，单击“通用选项卡”，在“启动对象”下拉列表框中，选择要设置为启动窗体的窗体，然后单击“确定”按钮即可。

4．添加工程

一个应用程序中可以包含多个工程文件，可以通过操作添加工程资料管理器窗口中的

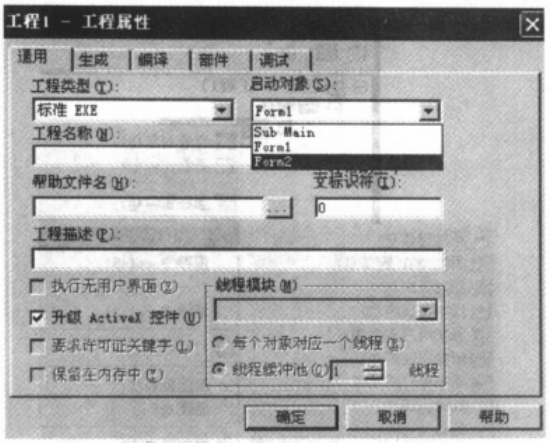


图 1-15 “工程属性”对话框

工程文件。如果添加“标准 EXE 工程”,可以通过标准工具栏中的“添加工程”按钮;如果添加其他类型的工程,可以通过“文件”菜单中的“添加工程”命令的对话框。

5. 删除工程

如果要删除工程,在工程管理器窗口中,右击要删除的工程,然后在弹出的快捷菜单中选择“移除工程”命令即可,如图 1-16 所示。

图 1-16 工程操作快捷菜单

6. 设置启动工程

一个应用程序可以包含多个工程,程序运行时执行的第一个工程为“启动工程”。一般情况下第一个建立的工程为默认的“启动工程”,如果要改变“启动工程”,可以在如图 1-16 所示的工程操作快捷菜单中选择“设置为启动工程”重新设置“启动工程”。

1.4.4 Visual Basic 6.0 程序设计步骤

一般来说,设计一个 Visual Basic 6.0 应用程序,需要以下几个步骤:

1. 新建工程

新建工程的默认名称为“工程 1”,其中包含一个默认的窗体“Form1”。可以通过下面的两种方法新建一个工程:

(1) 启动 Vsiual Basic 6.0 后,在“新建工程”对话框中选择“标准 EXE”,然后单击“打开”按钮。

(2) 单击 Vsiual Basic 6.0 的“文件”菜单中的“新建工程”命令,在“新建工程”对话框中选择“标准 EXE”,然后单击“打开”按钮。

2. 设计程序界面

设计程序界面的主要工作是向窗体设计器窗口中添加所需的控件,然后调整每个控件在窗体上的位置、大小等布局,以及对已添加到窗体中的控件进行复制、删除等编辑操作,这些操作可以通过菜单或者工具栏的操作方式完成。下面介绍设计程序界面中的基本操作。

(1) 添加控件。可以通过两种方法向窗体中添加控件:一种方法是单击控件工具箱中的控件图标,鼠标就会变成"十字"形状,这时在窗体中拖动鼠标就可以画出选择的控件;另一种方法是双击工具栏中的控件图标,该控件就会自动出现在窗体中。如果要单击一次控件图标就能画出多个相同的按钮,需要在单击控件图标时,同时按下"Ctrl"键。

(2) 调整控件。已添加到窗体中的控件,可以对其位置、大小进行调整。

(3) 复制和删除控件。添加到窗体中的控件可以对其进行复制、删除等操作。

(4) 调整界面布局。添加所需的控件,调整好其单个的位置和大小,并对窗体上所有控件的相对位置等整体布局进行调整,直到合适为止。

3. 设置对象的属性值

设置对象的属性值,可以通过两种方法:一种是在设计状态,通过其属性窗口设置对象的属性值;另一种是在程序运行状态,通过程序代码设置对象的属性值。如果在设计状态下设置对象的属性值,需要首先显示对象的属性列表,方法是选定窗体中要设置其属性值的对象,或者在属性窗口的对象下拉列表框中选择要设置属性值的对象名称。

4. 编写程序代码

Visual Basic程序设计的主要工作就是为对象的事件过程编写程序代码,编写程序代码需要进入代码窗口。方法是双击对象,进入代码编辑窗口;或者在工程资源管理器窗口中单击"查看代码"按钮,进入代码编辑窗口。

5. 获得MSDN帮助信息

在编写程序代码过程中,可以通过MSDN获得示例代码的帮助。用户可以通过"帮助"菜单,或者在代码编辑窗口中选定代码后按F1键,这两种方式均可获得帮助信息。

6. 运行和调试程序

如果程序没有语法错误,就可以在程序设计状态下,通过单击工具栏上的启动按钮或按F5键,就可以运行当前打开的工程。如果不能正常运行,就需要进行程序调试,从语法和算法等方面分析和查找错误,使其直至能正常运行。

7. 保存程序文件

保存程序时,程序中的每个工程文件、窗体文件及其他文件都要分别进行保存。如一个应用程序只包含一个工程,该工程中只包含一个窗体,第一次保存时,就会出现两次保存的对话框,第一次要求保存窗体文件,第二次要求保存工程文件。

8. 编译程序

编译程序就是把源程序文件编译成可执行文件,编译成功后的可执行文件,可以脱离Visual Basic 6.0集成开发环境而单独运行。

上述Visual Basic 6.0程序设计的一般步骤的顺序,并不是固定不变的,在程序设计过程中,有些步骤的顺序是可以调整的。

1.4.5 简单应用程序设计实例

下面以一个简单的应用程序设计为例,说明Visual Basic应用程序设计的一般步骤。

【例1-1】 如图1-17所示,程序运行时,在窗体标题栏上显示"简单程序示例",在窗体上显示"欢迎使用本系统",在窗体的下面有一个"退出"按钮,单击该"退出"按钮就会退出程序。下面按照程序设计的一般步骤,进行本题简单应用程序的设计。

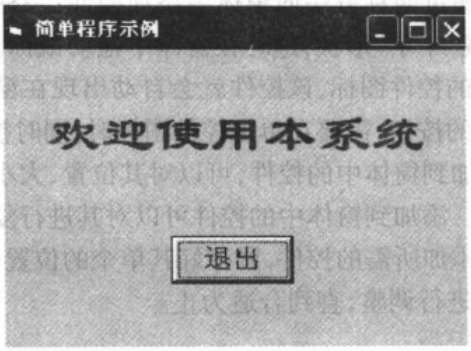


图 1-17 例 1-1 程序运行界面

1. 新建工程

新建一个“标准 EXE”工程，名称为工程 1，包含一个窗体，名称为 Form1，如图 1-18 所示。

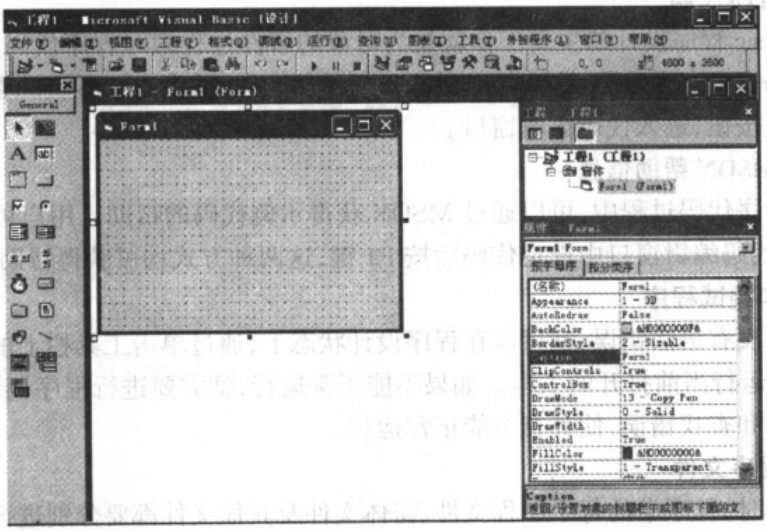


图 1-18 新建工程

2. 设计界面

如图 1-19 所示，在窗体中添加 1 个标签按钮，用来显示“欢迎使用本系统”；添加 1 个命令按钮，用来响应用户的单击操作。

(1) 单击工具箱中的标签(Label)控件图标，在窗体上拖动鼠标，画出 1 个标签控件，标签控件的名称为 Label1，标签上显示的文本也是 Label1。

(2) 双击工具箱中的命令按钮(CommandButton)控件图标，在窗体中添加 1 个命令按钮，名称为 Command1，按钮上显示的文本也是 Command1。

(3) 拖动鼠标分别选择调整每个控件的大小、位置及布局，选择 Label1 控件和 Command1 控件，使 Command1 控件为基准控件，单击“格式”菜单中“对齐”菜单项中的“居中对齐”命令，此时两个控件的中轴线垂直对齐。如图 1-19 所示。

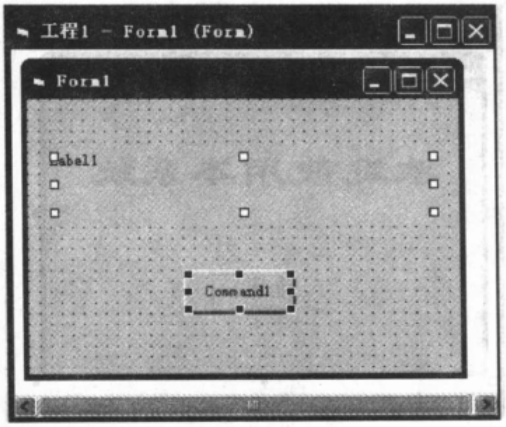


图 1－19　设计界面

3. 设置对象的属性值

窗体 Form1、标签 Label1、命令按钮 Command1 三个对象的属性设置值见表 1－1，按如下操作设置对象的属性值。

(1) 单击窗体 Form1，在属性窗口中，选择 Caption 属性，在属性值栏中，删除原来的取值 Form1，输入“简单程序示例”后回车，此时可以看到窗体标题栏中的文本已变成“简单程序示例”。如图 1－20 所示。

(2) 单击标签 Label1，在属性窗口中选择 Caption 属性，在属性值栏中，删除原来的取值 Label1，输入“欢迎使用本系统”后回车，此时可以看到标签中的文本已变成“欢迎使用本系统”；选择 Alignment 属性，选取“2－Center”属性值；选择 Font 属性，选取“隶书、常规、一号字”属性值；选择 ForeColor 属性，选取“&H000000FF&”属性值。如图 1－20 所示。

表 1－1　例 1－1 对象属性值

对象	属性	属性值
Form1	Caption	简单程序示例
Label1	Caption	欢迎使用本系统
	Alignment	2－Center
	Font	隶书、常规、一号字
	ForeColor	&H000000FF&
Command1	Caption	退出
	Font	宋体、常规、四号字

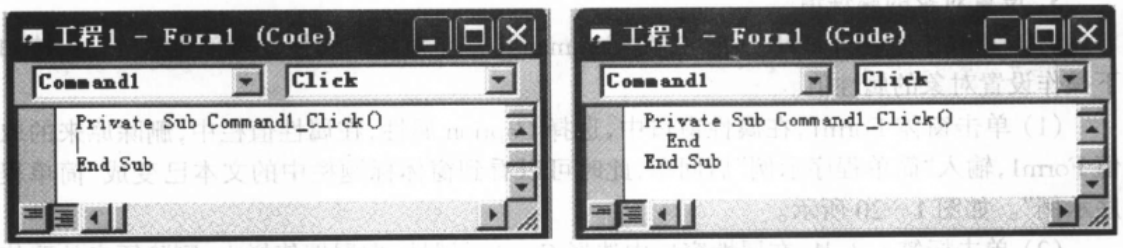
(3) 单击命令按钮 Command1，在属性窗口中选择 Caption 属性，在属性值栏中，删除原来的取值 Commandl，输入“退出”后回车，此时可以看到按钮中的文本已变成“退出”；选择 Font 属性，选取“宋体、常规、四号字”属性值；选择 ForeColor 属性，选取“&H000000FF&”属性值。如图 1－20 所示。

4. 编写程序代码

(1) 双击“退出”按钮，进入如图 1－21(a)所示的命令按钮单击事件代码窗口。

(2) 在图 1－21(a)的命令按钮单击事件代码窗口中，输入代码 End，如图 1－21(b)所示。

图 1－20 设置属性值



(a) (b)

图 1－21 编写程序代码

5. 获得 MSDN 帮助信息

如果不知道用什么语句来结束程序，可以利用“帮助”菜单打开 MSDN，“结束执行”为关键词进行索引，就会得到结束程序运行语句 End 的使用语法。如果知道结束程序运行的语句是 End，但是不知道其具体用法，可以在代码窗口中输入 End，然后选择 End，同时按 F1 键，就可以获得 End 语句的用法。如图 1－22 所示。

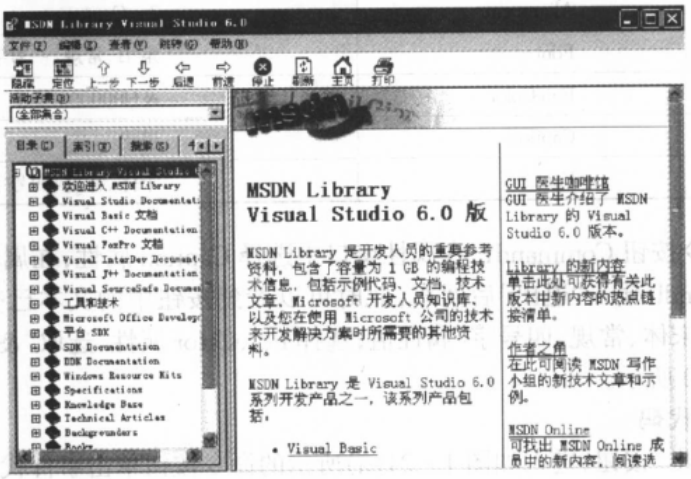


图 1－22 MSDN 浏览器窗口

6. 运行、调试程序

可以通过“运行”菜单中的“启动”命令、工具栏中的“启动”按钮、F5 键运行程序程序运行结果如图 1－20 所示。单击“退出”按钮，程序退出运行状态，回到设计状态。如果运行程序过程中出现错误，可以通过设置断点、单步运行等操作方法进行调试程序，使其能正常运行。

7. 保存程序文件

在桌面上建立一个文件夹，名称为“程序”，用来保存程序文件。单击工具栏中的“保存”按钮，在保存窗体对话框中，把原来默认的窗体文件名 Form1. frm 改成“程序示例. frm”，在保存工程对话框中，把原来默认的工程文件名工程 1. vbp 改成“程序示例. vbp”。

8. 生成可执行文件

单击“文件”菜单中的“生成工程 1. exe”，在弹出的“生成工程”对话框中，把原来默认的文件名工程 1. exe 改成“简单程序示例. exe”。

本书语法格式符号说明：

在介绍 Visual Basic 语法格式时，本书使用的符号含义如下：

(1) “< >”号，表示其中的参数为必选参数。

(2) “[ ]”号，表示其中的参数为可选参数，如果省略，则为默认值。

(3) “|”号，表示从其分隔的多个选项中选择一项。

(4) “,…”号，表示同类的项目重复多项。

(5) “…”号，表示省略了叙述中不涉及的语句部分。

习题一

1. 简答题

(1) 什么是程序？什么是算法？算法的描述方法有哪些？

(2) 程序设计语言分哪几种？它们的特点分别是什么？

(3) 如何理解面向对象的程序设计方法？

(4) 什么是对象的属性、方法、事件？

(5) Visual Basic 6.0 有哪些功能和特点？

(6) Visual Basic 6.0 的集成开发环境由哪些部分组成？

(7) Visual Basic 6.0 的程序管理结构是什么？

(8) 什么是“启动工程”和“启动窗体”？如何设置？

(9) Visual Basic 6.0 程序设计步骤有哪些？其中一些步骤的顺序是否可以改变？

2. 填空题

(1) Visual Basic 6.0 是面向(　　)的程序设计语言，采用(　　)驱动的编程机制。

(2) Visual Basic 6.0 提供了(　　)、(　　)和(　　)3 种版本，其中(　　)版本的功能最强。

(3) Visual Basic 6.0 有(　　)、(　　)和(　　)3 种工作状态。

(4) 安装了(　　)以后，才能使用 Visual Basic 6.0 的帮助文档。

(5) 保存由一个工程和一个窗体组成的 Visual Basic 6.0 应用程序时，需分别保存(　　)文件和(　　)文件。

(6) 窗体的(　　)属性，可以设置窗体标题栏中的内容。

(7) 标签控件的(　　)属性，决定其中文本的字体。

(8) Print 方法可以在(　　)、(　　)、(　　)和(　　)内输出数据,对应对象名称的关键字分别是(　　)、(　　)、(　　)和(　　)。

(9) 在 Print 方法中,(　　)号表示"紧凑格式"输出,(　　)号表示"松散格式"输出。

(10) Visual Basic 6.0 应用程序生成(　　)文件后,可以脱离 Visual Basic 6.0 集成开发环境运行。

3. 选择题

(1) 以下不属于 Visual Basic 系统的文件类型的是(　　)。

A. frm　　B. vbp　　C. doc　　D. vbg

(2) 当程序运行时,Visual Basic 系统自动执行启动窗体的(　　)事件过程。

A. Load　　B. Unload　　C. Click　　D. GotFocus

(3) ".bas"文件是 Visual Basic 的(　　)文件。

A. 窗体　　B. 工程　　C. 标准模块　　D. 类模块

(4) 为了使命令按钮(名称为 Command1)右移 200,应使用的语句是(　　)。

A. Command1. Move －200　　B. Command1. Move 200

C. Command1. Left = Command1. Left + 200　　D. Command1. Left = Command1. Left － 200

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(5) 在 Visual Basic 工程中,可以作为启动对象的程序是(　　)。

A. 任何窗体或标准模块　　B. 任何窗体或过程

C. Sub Main 过程或其他任何模块　　D. Sub Main 过程或任何窗体

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(6) 在代码编辑窗口中,当需要启动帮助系统时,按(　　)键,就会进入 MSDN Library 浏览器窗口。

A. ESC　　B. F1　　C. F5　　D. F10

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(7) 下列描述中,正确的是(　　)。

A. 程序就是软件

B. 软件开发不受计算机系统的限制

C. 软件既是逻辑实体,又是物理实体。

D. 软件是程序、数据与相关文档的集合。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(8) 下列叙述中正确的是(　　)。

A. 程序设计就是编制程序

B. 程序的测试必须由程序员自己去完成

C. 程序经调试改错后还应进行再测试

D. 程序经调试改错后不必进行再测试

【出处】2005 年 9 月全国计算机等级考试二级 Visual Basic

(9) 假定一个 Visual Basic 应用程序由一个窗体模块和一个标准模块构成。为了保存该应用程序,以下正确的操作是(　　)。

A. 只保存窗体模块文件

B. 分别保存窗体模块、标准模块和工程文件

C. 只保存窗体模块和标准模块文件

D. 只保存工程文件

【出处】2005 年 9 月全国计算机等级考试二级 Visual Basic

(10) 为了清除窗体上的一个控件,下列正确的操作是(　　)。

A. 按回车键

B. 按 Esc 键

C. 选择(单击)要清除的控件,然后按 Del 键

D. 选择(单击)要清除的控件,然后按回车键

【出处】2005 年 9 月全国计算机等级考试二级 Visual Basic

(11) 以下叙述中,错误的是(　　)。

A. 打开一个工程文件时,系统自动装入与该工程有关的窗体、标准模块等文件。

B. 当程序运行时,双击一个窗体,则触发该窗体的 DblClick 事件。

C. Visual Basic 应用程序只能以解释方式执行。

D. 事件可以由用户引发,也可以由系统引发。

【出处】2005 年 9 月全国计算机等级考试二级 Visual Basic

(12) 如果一个工程含有多个窗体及标准模块,则以下叙述中错误的是(　　)。

A. 任何时刻最多只有一个窗体是活动窗体。

B. 不能把标准模块设置为启动模块。

C. 用 Hide 方法只是隐藏一个窗体,不能从内存中清除该窗体。

D. 如果工程中含有 Sub Main 过程,则程序一定首先执行该过程。

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(13) 以下关于 Visual Basic 特点叙述中,错误的是(　　)。

A. Visual Basic 是采用事件驱动编程机制的语言。

B. Visual Basic 程序既可以编译运行,也可以解释运行。

C. 构成 Visual Basic 程序的多个过程没有固定的执行顺序。

D. Visual Basic 程序不是结构化程序,不具备结构化程序的三种基本结构。

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(14) 以下叙述中。错误的是(　　)。

A. 一个 Visual Basic 应用程序可以包含多个标准模块文件。

B. 一个 Visual Basic 工程可以含有多个窗体文件。

C. 标准模块文件可以属于某个指定的窗体文件。

D. 标准模块文件的扩展名是.bas。

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(15) 以下叙述中,错误的是(　　)。

A. 在 Visual Basic 中,对象所能响应的事件是由系统定义的。

B. 对象的任何属性既可以通过属性窗口设定,也可以通过成语语句设定。

C. Visual Basic 中允许不同对象使用相同名称的方法。

D. Visual Basic 中的对象具有自己的属性与方法。

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

4. 编程题

在窗体 Form1 上添加一个命令按钮 Command1 和一个标签 Label1,当程序运行后,单击该按钮时,在标签中显示“Visual Basic 程序设计”。

第 2 章　Visual Basic 程序设计基础

Visual Basic 程序设计的一个主要步骤和工作是编写程序代码，程序代码是由每一条语句组成的，语句是由数据、运算符、表达式等组成的。本章主要介绍 Visual Basic 语言的基本概念、基本数据类型、常量与变量、运算符与表达式、常用的内部函数及其应用。

2.1　Visual Basic 程序语言的基本概念

2.1.1　标识符

标识符是编程时为常量、变量、数据类型、过程、函数、类等定义的标识符号，即名字，利用它可以引用相应的常量、变量、数据类型、过程、函数、类。

关键字是 Visual Basic 语言中预先定义的作为程序中固有含义的标识符，也叫保留字，不能被重新定义。

Visual Basic 语言中所有标识符都有相同的命名规则，命名规则如下：

(1) 标识符必须以字母开头，后面可以跟字母、数字、下划线“_”，不能包含标点符号、空格等。

例如，6a、b. c、c d、d&e 等都是不合法的。

(2) 标识符的最大长度不能超过 255 个字符。

(3) 自定义的标识符不能与 Visual Basic 中的关键字同名。

例如，变量名不能用 True、End、If、Integer 等。

2.1.2　书写规范

Visual Basic 程序的基本单位是语句，编写程序代码时必须符合 Visual Basic 程序的语法规范。Visual Basic 程序的语句书写有如下规范和要求。

1. Visual Basic 程序代码中字母的大小写

(1) Visual Basic 程序代码中，不区分字母的大小写。

(2) Visual Basic 程序代码中关键字的首字母总是自动被转换成大写，其余字母被转换成小写。如果关键字是由英文单词组成的，那么自动将每个单词的首字母转换成大写。

例如，关键字 If、False、Long 等首字母均为大写字母。

(3) 对于用户自定义的变量、过程等名字，Visual Basic 以第一次定义时的大小写为准，以后每次再输入该变量、过程名字时，就会自动按第一次定义时的格式转换。

例如，第一次定义变量时的名字为 Student_Name，则以后不论使用大写，还是使用小写，当书写完该语句换行时，就会自动按 Student_Name 的大小写格式进行转换。

2. 语句按行书写

(1) Visual Basic 的语句按行书写，每行最多允许书写 255 个字符。

(2) Visual Basic 允许把多个语句合并到同一行中,但语句之间必须用冒号":"进行分隔。

例如,在一行中书写5个语句:

a=2:b=4:c=6:d=8:e=10

(3) 单行语句可以通过续行符"_"(由一个空字符和一个下划线字符组成),将一行语句分成若干行来书写。一行语句最多只能有25个续行。

例如,将一行语句分成两行书写:

```
Label1.Caption= _
"欢迎使用本系统!"
```

3.注释语句

程序代码中的注释语句用于在程序代码中添加注释。代码段中的注释语句并不参与程序的运行,只是起到提高程序可读性的作用,便于进行程序的调试和维护。Visual Basic 提供了两种方法来为程序代码添加注释。

(1) Rem 语句。语法如下:

```
Rem 注释文本
```

例如,添加注释,说明命令按钮 Command1 的 Click 事件的功能,书写注释如下:

```
Private Sub Command1_Click()
Rem 单击该按钮在标签中显示"欢迎您!"
```

或者在一行中书写注释如下:

```
Private Sub Command1_Click() :Rem 单击该按钮在标签中显示"欢迎您!"
```

(2) 单引号"'"注释符。语法如下:

```
'注释文本
```

使用单引号注释符"'"给代码添加注释,更加灵活,也更加常用。

例如:

```
Private Sub Command1_Click()
'单击该按钮在标签中显示"欢迎您!"
```

或者在一行中书写注释如下:

```
Private Sub Command1_Click()     '单击该按钮在标签中显示"欢迎您!"
```

2.2　数据类型

用于描述客观事物的数、字符,以及所有能输入到计算机中并被计算机程序加工处理的符号的集合统称为数据。数据是程序的必要组成部分,也是程序处理的对象,换言之,程序设计的主要功能就是进行各种数据的输入、处理、输出等。在高级语言中,"数据类型概念"体现数据结构的特点,数据类型不同,其占用的存储空间大小也不同。

可以从不同的角度对数据进行分类:

(1) 根据数据种类的不同,可以把数据划分为不同的类型,不同数据类型的数据处理方式也不相同。在 Visual Basic 中,提供了一些常用的基本数据类型,用户也可以在基本数据类型的基础上,根据具体需要定义新的数据类型。基本数据类型包括:Integer(整型)、Byte

(字节型)、String(字串符型)、Boolean(布尔型)、Date(日期型)、Object(对象型)、Variant(变体型)等,多达 12 种数据类型。

(2) 根据数据在程序运行期间是否发生变化,可将数据划分为常量、变量两种类型。

2.2.1 基本数据类型

不同类型的数据有不同的表示方法,不同类型的数据有不同的操作方式和不同的取值范围。Visual Basic 所提供的基本数据类型有 4 种,下面分别进行介绍。

1.数值型数据

Visual Basic 中的数值型数据可以分为整型和实型两大类。

(1) 整型数据是指不带有小数和指数符号的数,可分为 Integer(整型)、Long(长整型)、Byte(字节型)。

(2) 实型数据常指带有小数或指数符号的数,按其表现形式的不同可分为定点数和浮点数。常规表示的小数称为定点数,例如,2.71828、-3.1415926 等。浮点数由符号、尾数、指数三部分组成,例如,-1.732E+7,表示 -1.732×10^7。按其取值范围和精度的不同可以分为单精度实型(Single)、双精度实型(Double)、货币型(Currency)。

2.字符串型数据

字符串型数据(String)一般是指用于输出到屏幕或打印机上的字符或字符串,它是用双引号括起来的一串字符,根据其长度的确定与否可分为定长字符串和变长字符串两种。

3.布尔型数据

布尔型(Boolean)数据,也称逻辑型数据,它只有两个值:True(真)和 False(假),经常被用来表示逻辑判断结果。当把逻辑型数据转换为数值型时,True 成为 -1,False 成为 0;而把数值型数据转换为逻辑型时,0 转换为 False,而其他非 0 值转换为 True。

4.日期型数据

日期型(Date)数据用两个“#”符号把表示日期和时间的值括起来。例如,#10/17/2006#,#10/17/2006 22:21:00PM#。

Visual Basic 提供的基本数据类型见表 2-1。

表 2-1 Visual Basic 中的基本数据类型

基本数据类型	类型说明符	占用字节数	取值范围
Integer(整型)	%	2	-32768~32767
Long(长整型)	&	4	-2147482648~2147483647
Byte(字节型)		1	0~255
Single(单精度实型)	!	4	负数:-3.402823E+38~-1.401298E-45 正数:1.401298E-45~3.402823E+38
Double(双精度实型)	#	8	负数:-1.79769313486232E+308~-4.94065645841247E-324 正数:4.94065645841247E-324~1.79769313486232E+308
Currency(货币型)	@	8	-922337203685447.5808~922337203685447.5808
String(字符串型)	$	根据字符串的长短确定	
Boolean(布尔型)		2	True 或 False

续表 2-1

基本数据类型	类型说明符	占用字节数	取值范围
Date(日期型)		8	100.1.1~9999.12.31
Object(对象型)		4	
Variant(变体型)		不确定	任何数值,最大可达 Double 类型的范围

2.2.2 常量

计算机在处理程序时,对于输入的数据、参加运算的数据、运行结果等临时数据,必须将其装入计算机的内存中。在机器语言与汇编语言中,借助于内存单元的编号(地址)访问内存中的数据。而在高级语言中,需要将存放数据的内存单元命名,通过内存单元名来访问其中的数据,常量或变量本质上即为被命名的内存单元。在程序中,不同类型的数据能够以常量的形式出现,也能够以变量的形式出现。常量是指在程序运行过程中其值始终保持不变的数值、字符串等常数;变量是指在程序运行过程中其值可以发生变化的量。在 Visual Basic 程序中,有两种类型的常量,即直接常量和符号常量。直接常量是指直接表示的常量;符号常量是指用一个合法的符号(即标识符)代表常量,用来代替多次出现或难于记忆的常数值,提高代码的可读性和可维护性。

1.直接常量

直接常量根据数据类型可分为数值型常量、字符串型常量、布尔型常量、日期型常量 4 种。其中数值型常量又可分为整型常量和实型常量 2 种,整型常量包括整型常量和长整型常量 2 种;实型常量分为单精度浮点型常量、双精度浮点型常量、货币型常量 3 种。下面分别介绍这 8 种直接常量。

(1) 整型常量。即整数,根据不同进位计数制的表示形式,整型常量可表示为十进制、八进制、十六进制,见表 2-2。

表 2-2 Visual Basic 中的整型常量

类型	开头标志符号	结尾标志符号	可包含符号	取值范围
十进制			正、负号,0~8	-32768~32767
八进制	&O(字母 O)或 &		正、负号,0~7	-&O177777~&O177777
十六进制	&H 或 &h		正、负号,0~9,A~F 或 a~f	-&HFFFF~&HFFFF 或 -&hffff~&hffff

(2) 长整型常量。根据不同进位计数制的表示形式,长整型常量可表示为十进制、八进制、十六进制,见表 2-3。

表 2-3 Visual Basic 中的长整型常量

类型	开头标志符号	结尾标志符号	可包含符号	取值范围
十进制			正、负号,0~8	-2147483648~2147483647
八进制	&O(字母 O)或 &	&	正、负号,0~7	-&O37777777777~&O37777777777
十六进制	&H 或 &h	&	正、负号,0~9,A~F 或 a~f	-&HFFFFFFFF~&HFFFFFFFF 或 -&hffffffff~&hffffffff

(3) 实型常量。即实数,实型常量根据精度不同,可分为单精度实数、双精度实数。实

型常量由尾数、指数符号、指数三部分组成。其中尾数本身也是实数;单精度实数的指数符号为字母 E 或 e,双精度实数的指数符号为字母 D 或 d,含义是"乘以 10 的幂次";指数是整数。例如,2.7E+4! 表示单精度实型常量 2.7×10^4,-1.6d-4# 表示双精度实型常量 -1.6×10^{-4}。

(4) 货币型常量。货币型常量是一种特殊的实型常量,最多可精确到小数点后 4 位,用@符号作为标志符号,例如,39@、270.1234@等。

(5) 字符串型常量。字符串型常量要用双引号括起来,字符串的内容是指双引号内的若干个 ASCII 字符,不包括双引号本身。例如,"Good Morning!"、""等。

(6) 布尔型常量。也叫逻辑型常量,只有两种取值,即 True(真)和 False(假)。当布尔型常量转化为数值型常量时,True 转换成 -1,False 转换成 0;而当数值型常量转换为布尔型常量时,非 0 数值转换成 True,0 转换成 False。

(7) 日期型常量。日期型常量要用双"#"号括起来,双"#"号内包含合法的年月日时分秒(即日期常量)的内容。例如,#10/14/2006 11:20:30#、#10-14-2006#等。

2.符号常量

符号常量可分为两种,一种是用户定义的符号常量,另一种是系统内部定义的符号常量,即系统常量。

(1) 用户定义的符号常量。用户可以根据自己程序设计的需要定义符号常量,定义符号常量的语法格式为:

[Public|Private]Const 符号常量名[As 数据类型]=表达式[,常量名[As 数据类型]=表达式]…

其中:Public|Private 表示符号常量的有效范围,详见变量定义部分;常量必须是合法的标识符;一次可以定义多个符号常量,需用逗号隔开;符号常量的数据类型可以在 As 关键字后面进行说明,也可以在常量名的后面添加表 2-1 中的"类型说明符"来说明。例如:

```
Const PI As Double=3.1415926          '定义双精度符号常量 PI,代表常数 3.1415926
Const PI#=3.1415926                   '用类型说明符说明定义的符号常量 PI 的类型
Const PI3=PI*PI*PI                    '用已定义的符号常量定义新的符号常量
```

(2) 系统常量。系统常量是系统内部预先定义的,用户在编写代码时,不用说明即可引用,系统常量与应用程序的对象、方法、属性一起使用,一般以 Vb 为前缀。例如:VbRed 表示红色;VbSunday 表示星期日,其值为 1,Integer 类型,等等。要查看这些系统常量,单击菜单栏中"视图"菜单中的"对象浏览器"命令(或直接按"F2"键或直接按工具栏中的"对象浏览器"按钮),即可打开对象浏览器窗口,看到系统常量的名字。

2.2.3 变量

变量是指在程序运行过程中其值可以变化的量,实际上变量代表的是计算机中某种数据类型的存储单元。每个变量都有一个变量名,借助变量名就可以访问到内存中的数据。变量的两个最重要元素就是变量名和数据类型,变量名用于在程序中标识变量和使用变量的值,数据类型则确定变量中可以保存哪种类型的数据,变量所属的数据类型不同,其占用的内存存储空间的大小也不同。

1.变量的声明

变量的声明也称变量的定义,其实就是通知程序,按照变量的类型事先为其分配适当的

存储空间。在 Visual Basic 中，尽管不要求必须声明变量，但是要求读者应该培养良好的编程习惯，在使用变量前，应该先做变量声明。Visual Basic 中，变量的声明有显式声明和隐式声明两种方式。

(1) 显式声明。变量的显式声明就是在使用变量之前，使用 Dim、Private、Static、Public 语句声明一个或多个变量，前面的关键词分别表示局部的、私有的、静态的、公共的。变量声明的格式如下：

Dim|Private|Static|Public <变量名 1>[As <类型 1>][,<变量名 2>[As <类型 2>]]…

例如：

```
Dim a As Integer, b As Long          '定义 a 为整型变量,b 为长整型变量
```

在声明变量时，需要注意以下问题：

① Dim 为定义变量语句的关键字，也可以使用 Private、Static、Public 语句。

② 定义的变量名必须是合法的标识符，即符合标识符的命名规则。

③ 变量的数据类型在 As 后面进行说明，如果不进行说明，则变量为 Variant 类型；As 后面的数据类型表示该变量是可以存储的数据类型。

④ 可以一次声明多个变量，但要用逗号隔开。

⑤ Dim 和 Private 常用于在过程或模块中定义局部变量。当过程运行结束后，这些局部变量被释放；定义模块级变量时，Private 与 Dim 没有什么区别。

⑥ Static 用于在过程中定义静态变量，过程运行后变量不会被释放，将继续保留其值，下次可以再次引用。

⑦ Public 用于定义全局变量，该变量在整个程序中均有效，可以被应用程序的所有过程引用和访问。不能在过程中声明全局变量，只能在模块的声明段中声明全局变量。

此外，除了运用 Dim、Private、Static、Public 语句声明变量外，还可以通过在变量名后面后缀数据类型说明的方式，进行默认声明变量。例如，ab% = 9。

(2) 隐式声明。变量的隐式声明是指在程序中不对变量进行声明而直接引用，在使用前，此变量的类型为 Variant 类型。例如：

```
Mystring = "Visual Basic 程序设计"        'Mystring 为字符型变量
```

(3) 强制声明。尽管 Visual Basic 允许不定义变量就可以直接使用，但是如果把变量名书写错了，就会导致难以查找的问题。因此，在编程时应该使用 Visual Basic 提供的强制变量声明。强制变量声明有两种方法：

① 在类模块、窗体模块或标准模块的“通用|声明”部分添加下面的语句：

```
Option Explicit
```

② 通过“工具”下拉菜单中“选项”对话框中的“编辑器”选项卡，进行设置强制变量声明，如图 2-1 所示。

2. 变量的初始值与赋值

(1) 变量的初始值。在 Visual Basic 中，变量声明后，Visual Basic 就会给变量一个默认的初始值。变量的数据类型不同，赋给变量的初始值也不同。各种类型变量的初始值如下：

① 数值类型的变量，包括 Integer 类型、Long 类型、Single 类型、Double 类型、Byte 类型、Current 类型的变量，被声明后的初始默认值都是 0。

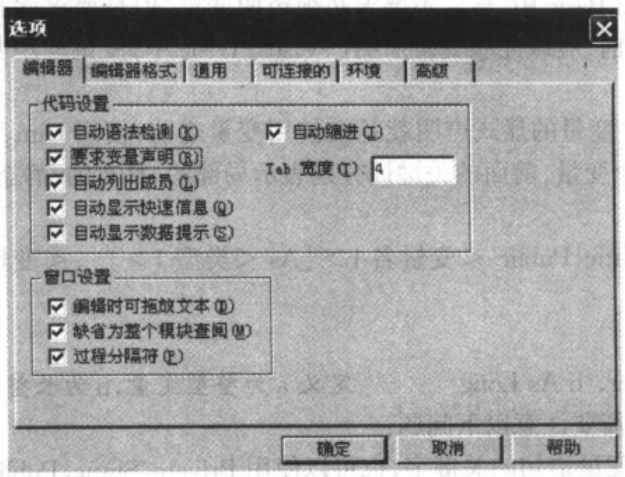


图 2-1 “强制变量声明”对话框设置

② String 类型的变量被声明后的初始默认值为""，即空字符串。

③ Date 类型的变量被声明后的初始默认值为 0:00:00。

④ Boolean 类型的变量被声明后的初始默认值为 False。

⑤ Variant 类型的变量被声明后的初始默认值为 Empty，即空值。

(2) 变量的赋值。在声明一个变量后，就可以给变量赋值。赋值语句用于将表达式的值赋给变量。赋值语句的格式为：

[Let]变量=表达式

例如：

```
Dim a As Integer
a=10:b=20:c=30
```

说明：

① Let 可以省略；“=”为赋值符号，不同于数学中等号的含义，表示把右边表达式的值赋给左边的变量。

② 只有当“=”右边表达式值的数据类型与左边变量的数据类型兼容时，该值才可以赋给变量，否则会强制将该值转换为变量的数据类型。

③ 当数据类型不匹配时，系统会提示出错。

④ 当数值型变量赋值超出其取值范围时，系统会提示溢出错误。

⑤ 当对定长字符型变量赋值时，如果字符串长度小于定长，则用空格将不足部分填满；如果字符串的长度太长，则截去超出部分的字符。

3. 变量的作用域

变量的作用域是指变量有效的作用范围，只有在有效范围内，变量才能被程序访问。根据变量作用范围的不同，可分为局部变量和全局变量。Visual Basic 允许在声明变量时指定它的作用范围，其范围可以是过程级、窗体级、模块级，或是全局变量。局部变量只能在声明它的过程(或模块)中有效，全局变量的作用范围最大，可以被所有过程和模块访问。变量的作用域大小取决于声明时的位置和关键字。Dim、Private、Static、Public 语句声明变量的作用域见表 2-4。

表 2-4　变量的作用域

范围类型	声明时所用关键字	声明位置	作用域	引用方式
过程级	Dim	过程内部	过程内部	变量名
	Static			
窗体级	Dim 或 Private	窗体的通用声明部分	窗体内部	变量名
	Public		本窗体或其他窗体	本窗体:变量名　其他窗体:窗体名.变量名
模块级	Dim 或 Private	模块的通用部分	模块内部	变量名
	Public		本模块或其他模块	本模块:变量名　其他模块:模块名.变量名

下面通过例题说明 Dim、Static、Private、Public 的作用范围。

【例 2-1】 用 Dim 和 Static 关键字分别定义一个过程级变量,在窗体单击事件中,单击一次窗体,变量的值将被加 1,对比一下,看用哪个关键字定义的变量能记住单击窗体的次数。程序运行界面如图 2-2 所示。

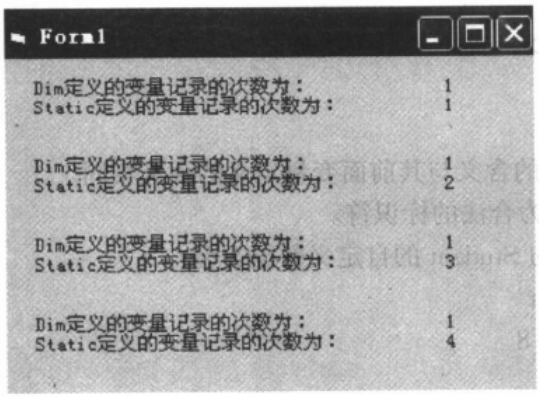


图 2-2　例 2-1 程序运行界面

程序设计步骤如下:

(1) 新建一个"标准 EXE"工程。

(2) 在窗体 Form1 的单击事件过程 Form_Click()中添加如下程序代码:

```
Option Explicit
Private Sub Form_Click()
  Dim countDim As Integer                    '用 Dim 定义的过程级变量
  Static countStatic As Integer              '用 Static 定义的过程级变量
  countDim = countDim + 1
  countStatic = countStatic + 1
  Print "Dim 定义的变量记录的次数为:  ", countDim
  Print "Static 定义的变量记录的次数为:  ", countStatic
  Print
End Sub
```

(3) 运行程序,单击 5 次窗体,观察程序运行结果,为什么两个变量记录单击窗体的次数不同。

关于用 Private 和 Public 关键字定义的变量作用范围的对比，读者可编写程序进行观察和分析。

2.2.4 自定义数据类型

如果在程序设计过程中，前面的基本数据类型不能满足编程的需要，还可以利用 Type 语句，在基本数据类型的基础上定义新的数据类型，称为用户自定义的数据类型，也叫做自定义数据类型。自定义数据类型与后面要介绍的枚举数据类型、数组也称为构造数据类型，也叫做复杂数据类型(相对于基本数据类型)。用户自定义数据类型可以包含多个元素，各元素可以具有不同的数据类型。

1.自定义数据类型的声明

自定义数据类型的声明格式为：

```
[Public|Private] Type <自定义数据类型名>
  <数据元素名 1>  As  <数据类型名>
  <数据元素名 2>  As  <数据类型名>
  ……
  <数据元素名 n>  As  <数据类型名>
End Type
```

说明：

(1) Public|Private 的含义与其前面变量定义的含义相同。

(2) 数据元素名应为合法的标识符。

例如，定义一个名为 Student 的自定义数据类型：

```
Public Type Student
  name As String * 8
  sex As String * 2
  age As Integer
  stature As Single
End Type
```

当然，也可以向前面符号常量的定义一样，使用定义的自定义数据类型，再定义新的自定义数据类型。例如，利用已定义的自定义数据类型 Student，再定义一个新的自定义数据类型 Student_Info，即学生信息。

```
Type Student_Info      '自定义 Student_Information 数据类型，省略时默认为 Public 类型
  Class As Sting
  Order As Integer
  Info As Student      '可以用已定义好的自定义数据类型 Student
  Address As String
End Type
```

2.自定义数据类型变量的声明

我们上面定义了一个新的数据类型 Student，它由 4 个元素(或成员)组成，分别用来表示学生的“姓名”、“性别”、“年龄”、“身高”。有了 Student 数据类型的定义，就可以用它来定义变量并访问它们。使用自定义数据类型的变量，可以通过“变量名.元素名”的方式引用。

例如：

```
Dim Student1 As Student,Student2 As Student
  Student1.name="刘翔"
  Student1.sex="男"
  Student1.age=19
  Student1.stature =1.79
```

注意：

(1) 引用自定义数据类型变量的元素时，使用“.”把变量名和元素连结起来。

(2) 自定义数据类型的声明不能放在过程中，而应放在窗体模块或标准模块中，在窗体模块中只能使用 Private 关键字。

2.2.5　枚举数据类型

如果一个变量只允许具有某几个可列举的取值情况下，可以采用枚举数据类型。所谓“枚举”就是指将变量可能的值一一列举出来，作为枚举数据类型的“成员”，变量的取值只限于这些成员所代表值的范围。在程序设计中，使用枚举数据类型的变量，有利于提高程序代码的可读性。

1.枚举数据类型的声明

声明枚举数据类型的一般格式为：

```
[Public|Private] Enum <枚举数据类型名>
  <成员名 1>  [<=常数表达式>]
  <成员名 2>  [<=常数表达式>]
  ……
  <成员名 n>  [<=常数表达式>]
End Enum
```

说明：

(1) 枚举数据类型的声明不能放在过程中，只能放在窗体模块、标准模块和类模块的声明部分中。

(2) Public|Private 的含义与其前面变量定义的含义相同，表示该枚举数据类型的有效范围，默认的情况下，枚举数据类型被定义为 Public 类型。

(3) “枚举名”应是一个合法的标识符。

(4) “常数表达式”表示枚举数据类型中各个成员的值，类型为 Long；它是可选的，如果省略常数表达式，则成员的值为默认值，即成员 1 的值为 0，后面的依次加 1。

例如，定义一个表示交通岗红绿灯的枚举数据类型 Traffic：

```
Enum Traffic
  RedLamp=1
  GreenLap=2
  AmberLamp=3
End Enum
```

2.枚举数据类型变量的声明

定义一个枚举数据类型后，就可以利用已定义的枚举数据类型进行枚举数据类型变量

的声明。其语法格式与声明普通变量相同。例如，可以利用上面定义的枚举数据类型 Traffic，声明一个该数据类型的变量，并给该变量赋值：

```
Dim Traf1 As Traffic          '声明一个 Traffic 类型的变量 Traf1
Traf1 = RedLamp               'Traf1 的值为 1
```

2.3 运算符与表达式

运算（即操作）是对数据进行处理的过程，最基本的运算可以用简洁的符号来描述，这些代表一定运算功能的符号即运算符，可以对常量、变量（即操作数）进行处理。Visual Basic 中的运算符分为算数运算符、关系运算符、逻辑运算符、字符串连接运算符 4 种。由各种运算符与运算数（常量、变量、函数、对象等）组成的式子称为表达式，Visual Basic 中的表达式与运算符对应分为算数表达式、关系表达式、逻辑表达式、字符串表达式 4 种。

2.3.1 算数运算符与算数表达式

算数运算符是用来进行数值运算的运算符，Visual Basic 提供了 8 种算数运算符。由算数运算符与运算数连接起来的式子称为算数表达式。算数运算符与算数表达式见表 2－5。

表 2－5 算数运算符与算数表达式

运算符	运算关系	算数表达式实例	表达式结果（设 a＝5，b＝2）
＋	加	a＋b	7
－	减	a－b	3
*	乘	a * b	10
/	除	a/b	2.5
∧	乘方	a∧b	25
\	整除	a\b	2
Mod	求余数（取模）	a Mod b	1
－	取负	－a	－5

以下几点应注意：

（1）算术运算符两边的操作数应是数值型，如果是字符或布尔型，则自动转换成数值类型后再运算。

（2）除（/）运算，运算数可以为整数或者浮点数，其运算结果的类型由其值决定。

（3）整除（\）运算，运算数一般为整型数，当运算数含有小数时，则先按照“四舍五入”的规则转换成整数，然后再运算，结果为整型数。

（4）取余（Mod）运算，如果操作数为实数，首先对其进行“四舍五入”取整，然后再求模，运算结果的符号取决于左边的操作数。

2.3.2 关系运算符与关系表达式

关系运算符也称比较运算符，用于对两个表达式值之间的关系进行对比，Visual Basic 提供 8 种关系运算符。由关系运算符与运算数组成的式子，称为关系表达式，对应于关系运算符，也有 8 种关系表达式。关系表达式的运算结果是一个逻辑值，即有两种情况，当其关系成立时，关系表达式的值为 True；其关系不成立时，表达式的值为 False。关系运算符与关

系表达式见表 2－6。

表 2－6　关系运算符与关系表达式

运算符	运算关系	关系表达式实例	表达式结果(设 a=5,b=2)
=	等于	a=b	False
<>或><	不等于	a<>b	True
>	大于	a>b	True
>=	大于或等于	a>=b	True
<	小于	"a" < "b"	True
<=	小于或等于	a<=b	False
Like	字符串匹配	"a" Like "ab"	True
Is	对象引用比较	Command1 Is Command2	False

说明：

(1) 关系运算符两边的值或表达式的类型应一致。

(2) 如果两个操作数是数值型，则按其大小进行比较。

(3) 如果两个操作数是字符型，按其 ASCII 码值进行比较。在比较两个字符串时，首先从比较两个字符串的第一个字母开始，然后比较第二个，依次类推。

(4) 如果两个操作数是汉字字符，则按其拼音顺序进行比较，汉字字符大于西文字符。

(5) 数学不等式 3≤a≤b，在 Visual Basic 中不能写成 3<=a<=b，因为数学中不成立，但在 Visual Basic 中，该关系表达式相当于(3<=a)<=b，其值为 True，不能正确表达数与不等式的结果。在 Visual Basic 中，应使用逻辑表达式处理上式。

2.3.3　逻辑运算符与逻辑表达式

逻辑运算也称为布尔运算，用来判断操作数之间的逻辑关系，Visual Basic 中提供 6 种逻辑运算符。由逻辑运算符与操作数组成的式子称为逻辑表达式，对应逻辑运算符，Visual Basic 提供了 6 种逻辑表达式，见表 2－7。

表 2－7　逻辑运算符与逻辑表达式

x	y	Not x	x And y	x Or y	x Xor y	x Eqv y	x Imp y
True	True	False	True	True	False	True	True
True	False	False	False	True	True	False	False
False	True	True	False	True	True	False	True
False	False	True	False	False	False	True	True

说明：

(1) 逻辑运算符中，Not(非)、And(与)、Or(或)最为常用。其中 And(与)、Or(或)经常用于多个关系表达式的逻辑运算，此时，And(与)运算必须在多个条件同时全部为 True 时的结果才为 True，而 Or(或)运算只要多个条件中有一个条件为 True 时的结果就为 True。

(2) 如果运算数为数值数据，则先将数值数据转换为二进制数(补码形式)进行按位运算，此时 1 为 True，0 为 False，然后再把二进制结果转换成原运算数的数据类型。最后按照

非 0 值为 True,0 值为 False 的对应关系得出逻辑运算结果。Visual Basic 中一般以 -1 表示 True,以 0 表示 False。

2.3.4 字符串运算符与字符串表达式

字符串连接运算是把两个字符串连接成一个新的字符串,Visual Basic 提供 2 种字符串连接运算符,即"&"和"+"。由字符串运算符和字符串组成的式子,称为字符串表达式,与字符串运算符对应,字符串表达式也有 2 种。见表 2-8。

表 2-8 字符串运算符与字符串表达式

运算符	运算关系	表达式实例	表达式结果
&	连接	"Mp" & "4"	"Mp4"
		"Mp" & 4	"Mp4"
+	连接	"Mp" + "4"	"Mp4"
		"Mp" + 4	结果出错(类型不匹配)

说明:

(1) "&"和"+"两种字符串运算符在进行字符串连接运算时是等价的,二者的区别在于"&"可以将非字符串类型的数据转换成字符串后再进行连接运算,而"+"运算符不能自动转换,会由于类型不匹配导致运算结果出错。

(2) 如果两个操作数都是字符串类型,则结果也是字符串类型。

(3) 如果两个操作数都是 Null 时,则结果也是 Null。

(4) 当只有一个操作数是 Null 时,则将其作为长度的字符串,即空字符串(" ")。

(5) Empty 作为长度的字符串时,即空字符串(" ")。

2.4 运算符的优先级

运算符的优先级即各种运算的优先运算顺序。在一个表达式中同时出现多种运算符时,按照运算符的优先级来决定运算的先后次数,即优先级高的运算符将优先得到处理。4 类运算符优先级由高到低的次序为算数运算符、字符串连接运算符、关系运算符、逻辑运算符。详见表 2-9 所示,表中优先级的数字越小,表示优先级越高。

表 2-9 运算符的优先级

运算符种类	优先级	运算符
算数运算符	1	^
	2	—(取负)
	3	* /
	4	\
	5	Mod
	6	+ —
字符串运算符	7	& +(字符串连接符)
关系运算符	8	= > < <> >= <=

续表 2-9

运算符种类	优先级	运算符
逻辑运算符	9	Not
	10	And
	11	Or　Xor
	12	Eqv
	13	Imp
其　他	14	=(赋值符号)

下面根据运算符的优先级,计算一个表达式的值。例如,当 a=1,b=2,c=3,d=4 时,表达式 a+b>c+d>=d∧b\d/b*c And a-b<=c-d Or Not d>0 Or d 的值为 4,相当于计算表达式(((a+b)>(c+d)>=(d∧b))\(d/b*c))And ((a-b)<=(c-d))Or Not (d>0)Or d 的值。

说明:

(1) 表达式中的括号最优先,相同优先级的运算按照从左到右的顺序进行。

(2) 当运算数都是 Variant 类型,则运算结果为 Boolean 类型。

(3) 当两个运算数的数据类型的存储长度不同时,运算结果的数据类型为存储长度较长的数据类型。

(4) 除法运算不论运算数是哪种数据类型,运算结果的数据类型都是 Double 类型。

2.5　常用内部函数

函数是一种特定的运算,在程序中要使用一个函数时,只要给出函数名,并给出一个或多个参数,就能得到它的函数值。Visual Basic 的函数分为内部函数和用户自定义函数。用户自定义函数是由用户根据自己的需要定义的函数(见第 6 章);内部函数也叫标准函数或公共函数,是由 Visual Basic 系统提供的。下面介绍几种常用的内部函数,包括数学函数、字符串函数、数据类型转换函数、日期和时间函数、随机函数、输入输出函数等。

2.5.1　数学函数

数学函数主要是用来进行各种数学运算。现介绍 11 种常用的数学函数,包括三角函数、绝对值函数、符号函数、平方根函数、指数函数、对数函数、取整函数等。见表 2-10。

表 2-10　数学函数

函数名	功能说明	应用实例	结果(函数值)
sin(x)	求正弦函数	sin(1)	0.479425538604203
cos(x)	求余弦函数	cos(x)	0.877582561890373
tan(x)	求正切函数	tan(x)	0.54630248984379
atn(x)	求反正切函数	atn(x)	0.463647609000806
Abs(x)	求绝对值	Abs(-9)	9
Sgn(x)	求数字的符号,x 为正数时结果为 1,x 为负数时结果为 -1,x 为 0 时结果为 0	Sgn(-9)	-1

续表 2－10

函数名	功能说明	应用实例	结果(函数值)
Sqr(x)	求平方根	Sqr(9)	3
Exp(x)	求指数函数 e^x	Exp(2)	7.38905609893065
Log(x)	求自然对数 Ln(x)	Log(2)	0.693147180559945
Int(x)	求不大于 x 的最大整数	Int(—9.5)	—10
Fix(x)	取整,求 x 的整数部分	Fix(—9.5)	—9

2.5.2 字符串函数

字符串函数主要用来对字符串进行处理,包括在一个字符串中截取子串、去除字符串中的空格、字符串字母的大小写转换,等等。常用的 15 种字符串函数见表 2－11。

表 2－11 字符串函数

函数名	功能说明	应用实例	结果(函数值)
Left(字符串,n)	从字符串的左端截取 n 个字符,结果为截取的子串	Left("abcd1234",4)	"abcd"
Right(字符串,n)	从字符串的右端截取 n 个字符,结果为截取的子串	Right("abcd1234",4)	"1234"
Mid(字符串,n,m)	从字符串左端的第 n 个字符开始截取 m 个字符,结果为截取的子串	Mid("abcd1234",3,5)	"cd123"
Ltrim(字符串)	删除字符串左端的空格	Ltrim("abcd1234")	"abcd1234"
Rtrim(字符串)	删除字符串右端的空格	Rtrim("abcd1234")	"abcd1234"
Trim(字符串)	删除字符串两端的空格	Ttrim("abcd1234")	"abcd1234"
Space(n)	产生由 n 个空格组成的字符串	"abcd"& Space(3) & "1234"	"abcd 1234"
Len(字符串)	求字符串的长度	Len("abcd1234")	8
InStr([m,]字符串 1,字符串 2[,n])	在字符串 1 中查找字符串 2 第一次出现的位置	InStr("abcd1234","123")	5
Lcase(字符串)	将字符串的大写字母转换成小写字母	Lcase("ABcd1234")	"abcd1234"
Ucase(字符串)	将字符串的小写字母转换成大写字母	Ucase("ABcd1234")	"ABCD1234"
Val(字符串)	把字符串转换为数值型数据	Val("1234. 5678")	1234.5678
Str(数值)	把数值转换为字符	Str(1234.5678)	"1234.5678"
Chr(数值)	求以数值为 ASCII 码的字符	Chr(65)	A
Asc(字符串)	求字符串首字母的 ASCII 码值	Asc("abcd1234")	97

说明:

(1) 表 2－11 的"结果(函数值)"一栏中的双引号用于说明函数值为字符型数据,实际上程序运行的结果中没有双引号。

(2) 在 InStr 函数中,可选参数 m 表示比较类型,由 m 的取值来确定比较时是否区分字母的大小写。当 m＝0 时,表示区分字母的大小写;m＝1 时,表示不区分字母的大小写,如果省略该参数项,系统缺省值为 0。

(3) 在 Left 函数和 Right 函数中,如果 n 值大于或等于字符串长度,则取整个字符串;在 Mid 函数中,如果 n 值大于字符串长度,则返回空字符串,如果省略 n 选项,则取出从首位置开始的所有字符。

2.5.3 数据类型转换函数

数据类型转换函数主要用于在不同的数据类型之间进行转换。常用的 8 种数据类型转换函数见表 2-12。

表 2-12 数据类型转换函数

函数名	功能说明	应用实例	结果(函数值)
CInt(数值表达式)	将数值表达式的值转换为 Integer 类型,若表达式的值为小数,则四舍五入	CInt(10.51)	11
CLng(数值表达式)	将数值表达式的值转换为 Long 类型,若表达式的值为小数,则四舍五入	CLng(10.51)	11
CSng(数值表达式)	将数值表达式的值转换为 Single 类型	CSng(10.51)	10.51
CDbl(数值表达式)	将数值表达式的值转换为 Double 类型	CDbl(10.51)	10.51
CStr(表达式)	将数值表达式的值转换为 String 类型	CStr(10.52)	"10.51"
CDate(表达式)	将数值表达式的值转换为 Date 类型	CDate(10.51)	1900-1-9 12:14:24
CVar(数值表达式)	将数值表达式的值转换为 Variant 类型	CVar(10.51)	10.51
CCur(数值表达式)	将数值表达式的值转换为 Currency 类型,若表达式值的小数位大于 4 位,则在第 5 位四舍五入	CCur(10.12345)	10.1235

说明:表 2-12 的"结果(函数值)"一栏中的双引号表示字符串数据类型,实际上程序运行结果没有双引号。

2.5.4 日期和时间函数

VisualBasic 提供的日期和时间函数主要用于对日期和时间进行操作。常用的 10 种日期和时间函数见表 2-13。

表 2-13 日期和时间函数

函数名	功能说明	应用实例	结果(函数值)
Date 或 Date()	返回系统当前日期字符串,格式为 yy-mm-dd	Date 或 Date()	2006-10-3
Yaer(日期)	求日期字符串中的年份,结果为 Integer 类型	Year(#10/3/2006#)	2006
Month(日期)	求日期字符串中的月份,结果为 Integer 类型	Month(#10/3/2006#)	10
Day(日期)	求日期字符串中的日子,结果为 Integer 类型	Day(#10/3/2006#)	3
Weekday(日期)	求日期中的星期,结果为 Integer 类型 1 代表星期日,7 代表星期六	Weekday(#10/3/2006#)	3
Now 或 Now()	返回系统当前日期和时间字符串,格式为 yy-mm-dd hh:mm:ss	Now 或 Now( )	2006-10-3 10:32:55
Hour(时间)	求时间中的小时,结果为 Integer 类型	Hour(#15:32:55#)	10
Minute(时间)	求时间中的分钟,结果为 Integer 类型	Minute(#15:32:55#)	32
Second(时间)	求时间中的秒,结果为 Integer 类型	Second(#15:32:55#)	55

2.5.5 随机函数

随机函数用来产生一个随机数，在测试、模拟和游戏程序中，经常要用到随机数。在随机数产生时，需要一个随机种子，随机种子不同，产生的随机数也不同。Visual Basic 提供的随机函数为 Rnd(x)，可以产生[0,1]之间的单精度随机数，见表 2－14。

表 2－14 随机函数

函数名	功能说明	应用实例	结果(函数值)
Rnd 或 Rnd(正数)	以上一个随机数作为本次产生随机数的种子	Print Rnd Print Rnd Print Rnd(10) Print Rnd(10)	.6055475 .533424 .5795186 .2895625
Rnd(负数)	该负数相同，产生的随机数就相同	Print Rnd(－10) Print Rnd(－10) Print Rnd(－20) Print Rnd(－20)	.3276443 .3276443 .8276443 .8276443
Rnd(0)	产生与上一个随机数相同的随机数	Print Rnd Print Rnd(0)	.1518518 .1518518
Randomize[(n)]	使 Rnd(x)每次产生的随机数不同	Randomize Print Rnd Print Rnd	第 1 次 .6871759 .7760388 第 2 次 .9505426 .6734381

说明：

(1) Int(Rnd * n)产生 0,1,…,n 中的随机整数(n 为整数)。

(2) Int(Rnd * n)＋1 产生 1,2,…,n－1 中的随机整数(n 为整数)。

(3) Int(Rnd * (b－a＋1)＋a)产生[a,b]之间的随机整数。

(4) Chr(Int(Rnd * 26)＋65)随机产生一个大写英文字母。

(5) Chr(Int(Rnd * 26)＋97)随机产生一个小大写英文字母。

2.5.6 输入输出函数

输入输出函数主要用来实现数据的输入和输出，在 Visual Basic 提供的输入输出函数中，常用的有 Format 函数、Tab 函数、InputBox 函数、MsgBox 函数。其中 Format 函数、Tab 函数用于在 Print 方法中实现对输出数据格式的控制，InputBox 函数用来产生输入对话框，MsgBox 函数用来产生输出对话框。下面进行分别介绍。

1. Format 格式输出函数

在 Print 方法中使用 Format 函数，可以使数值型、日期型、字符型数据按照指定的格式进行输出，该函数的使用格式如下：

Format(＜表达式＞,[＜格式字符串＞])

例如：

```
Print Format(1234.5678,"####.###")   '输出 1234.568,#号为占位符
```

格式中的表达式可以是常量或表达式；格式字符串可以是字符串常量或字符串变量，一

般由专用的格式说明符组成。常用的格式说明符见表 2－15。

表 2－15　格式说明符

格式说明符	功能说明	举例	输出格式
#	数字占位符，输出一个数字或什么都不输出，前后不补 0	Format(123.456,"####.##")	123.46
0	数字占位符，输出一个数字，前后补 0	Format(123.456,"0000.00")	0123.46
.	小数点占位符	Format(1234.56,"00,000.00")	01,234.56
,	千位分隔符占位符	Format(1234.56,"#,####.##")	1,234.56
%	百分比符号占位符，表达式的值自动乘以 100	Format(0.123456,"###.##%")	12.35%
$	美元符号占位符	Format(123.456,"$##,###.##")	$123.46
+，−	正、负号占位符	Format(123.456,"+####.##")	+123.46
E+，E−	指数符号占位符	Format(123.456,"0.00E+00")	1.23E+02
@	字符占位符，输出字符或输出空白	Format(123.456,"@@@@@@@@@@")	123.456
ttttt	用 5 个 t 表示将时间按完整的时、分、秒格式输出	Format(Time,"ttttt")	18:59:05
dddddd	用 6 个 d 表示将日期按年、月、日格式输出	Format(Date,"dddddd")	2006 年 10 月 3 日
<	将表达式中的大写字母转换为小写字母输出	Format("ABC123","<")	abc123
>	将表达式中的小写字母转换为大写字母输出	Format("abc123",">")	ABC123

2. Tab 格式输出函数

在 Print 方法中，要把一个将输出的内容定位在第 n 列上，可以使用 Tab 函数。例如：

Print "学号";Tab(20);"姓名";Tab(40);"性别"

输出结果如下：

学号　　姓名　　性别

上面的输出结果中，“学号”在第 1 列输出，“姓名”在第 20 列输出，“性别”在第 40 列输出。Tab(n)函数与输出内容之间用分号进行分隔。

3. InputBox 函数

InputBox 函数用来产生一个输入对话框(图 2－3)，该函数的格式为：

InputBox("提示文本"[,"对话框标题"][,"默认值"])

例如，利用下面的代码，产生一个如图 2－3 所示的输入对话框。

```
Dim name As String
name = InputBox("请按默认格式输入您的姓名","姓名","HONG Zhan－hui")
```

说明：

(1) “提示文本”是一个字符串，用来显示对话框内的提示信息。如果提示信息较多，需要多行输出，可以在各行之间使用回车符 Chr(13)或者换行符 Chr(10)。

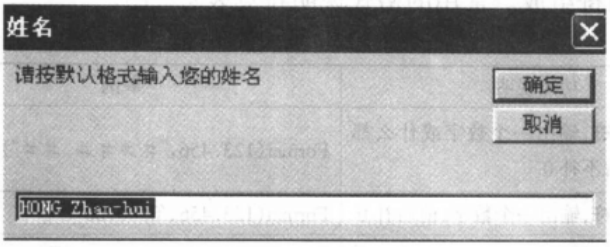


图 2-3 输入对话框

(2)“对话框标题”是一个字符串,用来设置对话框标题栏的内容,如果省略此项,标题栏上显示当前工程的名称。

(3)“默认值”是一个字符串,用来设置对话框内输入文本框的默认内容。如果省略此项,则输入文本框为空白。

(4) InputBox 函数执行后,其返回的函数值是一个字符串。当输入完内容后,如果单击“确定”按钮,则函数值为用户输入的内容;如果单击“取消”按钮,则返回一个空字符串""。在上例中,InputBox 函数的返回值被保存在变量 name 中。

(5) InputBox 如果不用括号,也可以当作一条语句使用,与函数的效果相同。

4. MsgBox 函数

MsgBox 函数用来产生一个消息输出对话框(图 2-4),给用户提供操作信息,MsgBox 函数的参数见表 2-16。该函数的语法格式为:

MsgBox("提示文本"[,按键钮组合 + 图标类型 + 默认按钮][,对话框标题])

例如,利用 MsgBox 函数产生一个如图 2-4 所示的消息输出对话框。

```
Dim wrong As Integer
Wrong = MsgBox("驱动器中数据读取出现错误!", vbAbortRetryIgnore + vbCritical + _
vbDefaultButton1,"错误提示")
```

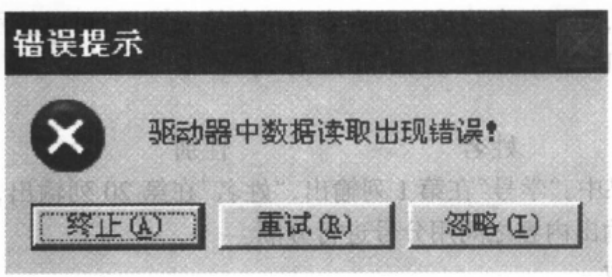


图 2-4 消息输出对话框

说明:

(1)“提示文本”是一个字符串,用来显示输出对话框内的提示信息。

(2)“按钮组合 + 图标类型 + 默认按钮”是 3 个数值型常量的组合,它们用“+”号分隔,每一项均可省略。其中“按钮组合”描述的是消息框中显示按钮的数目和类型;“图标类型”描述图标的样式;“默认按钮”确定对话框中的默认按钮,默认按钮表面有虚线框,当按下回车键时与单击该按钮的效果相同。

(3)“对话框标题”为可选项,表示对话框标题的内容,默认值为当前工程的名称。

(4) MsgBox 函数的返回值是一个整型常量,表示用户按下不同的按钮,见表 2－17。

(5) MsgBox 函数如果其参数不带括号,也叫 MsgBox 语句,与函数的效果相同,只不过是不能返回函数值。

表 2－16　MsgBox 函数的参数

参数	符号常量	数值	说　明
组合按钮	vbOkOnly	0	只显示“确定”按钮
	vbOkCancel	1	显示“确定”、“取消”2 个按钮
	vbAbortRetryIgnore	2	显示“终止”、“重试”、“忽略”3 个按钮
	vbYesNoCancel	3	显示“是”、“否”、“取消”3 个按钮
	vbYesNo	4	显示“是”、“否”2 个按钮
	vbRetryCancel	5	显示“重试”、“取消”2 个按钮
图标类型	vbCritical	16	显示 Critical 图标
	vbQuestion	32	显示 Question 图标
	vbExclamtion	48	显示 Exclamation 图标
	vbInformation	64	显示 Information 图标
默认按钮	vbDefaultButton1	0	第 1 个按钮为默认按钮
	vbDefaultButton2	256	第 2 个按钮为默认按钮
	vbDefaultButton3	512	第 3 个按钮为默认按钮
	vbDefaultButton4	768	第 4 个按钮为默认按钮

表 2－17　MsgBox 函数的参数

符号常量	直接常量	含　义
vbOk	1	单击了“确定”按钮
vbCancel	2	单击了“取消”按钮
vbAbort	3	单击了“终止”按钮
vbRetry	4	单击了“重试”按钮
vbIgnore	5	单击了“忽略”按钮
vbYes	6	单击了“是”按钮
vbNo	7	单击了“否”按钮

2.6　应用实例

【例 2－2】 设计一个程序,当程序运行后单击窗体,弹出如图 2－5(a)所示的输入对话框,输入姓名,单击“确定”按钮后弹出如图 2－5(b)所示的输入对话框,输入性别;单击“确定”按钮后弹出如图 2－5(c)所示的输入对话框,输入生日;单击“确定按钮后弹出如图 2－5(d)所示的输出对话框,提示用户的姓名、性别、年龄基本信息。

程序设计步骤如下:

(1) 新建一个“标准 EXE”工程。

(2) 在窗体 Form1 的 Click 事件过程 Form_Click()中添加如下程序代码:

```
Option Explicit
Private Sub Form_Click()
```

```
    Dim name As String                                    '用来保存用户的姓名
    Dim sex As String                                     '用来保存用户输入的性别
    Dim birString As String                               '用来保存用户输入的生日
    Dim birDate As Date                                   '用来保存日期型的生日
    Dim age As Integer                                    '用来保存年龄
    name = InputBox("输入姓名","name")                    '生成输入姓名的对话框
    sex = InputBox("输入性别","sex")                      '生成输入性别的对话框
    birString = InputBox("输入生日","birthday","1986-9-25")  '生成输入生日的对话框
    birDate = CDate(birString)                            '把生日转换成日期型
    age = Year(Date) - Year(birDate)                      '计算年龄
    MsgBox name & Space(2) & sex & Space(2) & age & "岁",vbInformation, _
    "信息提示"                                            '生成信息提示输出对话框
End Sub
```

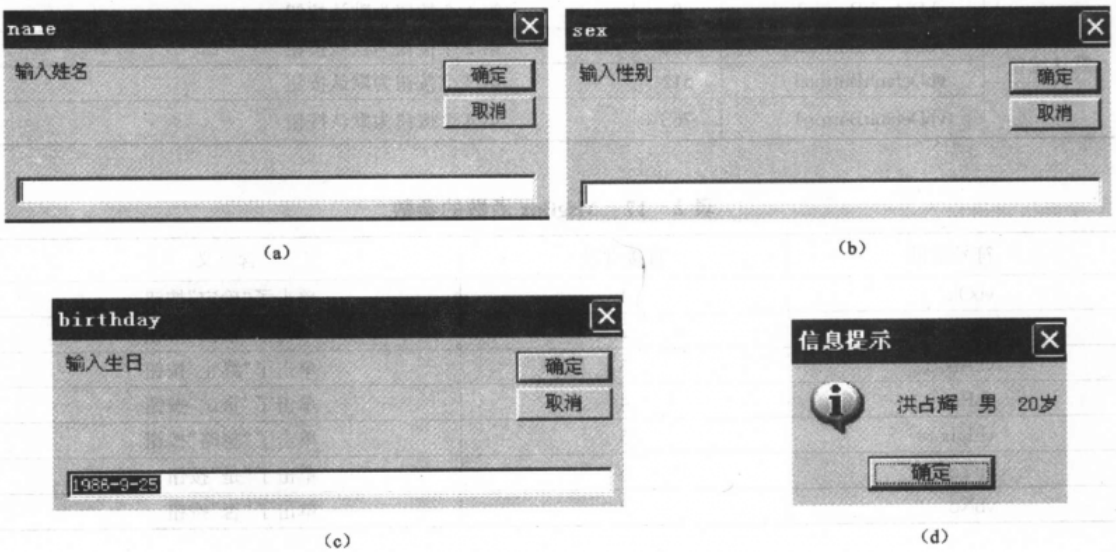


图 2-5 输入对话框

(a) 输入姓名;(b) 输入性别;(c) 输入生日;(d) 输入信息

(3) 运行程序,输入姓名、性别、生日,出现用户基本信息的提示。

习题二

1. 简答题

(1) 什么是标识符? 标识符的命令规则有哪些?

(2) Visual Basic 6.0 的基本数据类型包括哪几种?

(3) 什么是常量? 什么是变量? 举例说明如何定义一个变量?

(4) Visual Basic 6.0 的运算符有哪些? 它们的优先级是什么?

(5) Visual Basic 6.0 的表达式有哪些?

(6) Visual Basic 6.0的常用内部函数有哪些?

2.填空题

(1) 与数学表达式 a≤x≤b 对应的 Visual Basic 表达式是(　　)。

(2) 语句 Print Int(12345.6789 * 100 + 0.5)/100 的输出结果是(　　)。

【出处】2005年4月全国计算机等级考试二级 Visual Basic

(3) 可以得到[10,20]之间的一个随机整数的 Visual Basic 表达式是(　　)。

(4) 声明一个静态变量的关键字是(　　),声明一个全局变量的关键字是(　　)。

(5) 将 PI 定义为符号常量并赋值为 3.1415926 的 Visual Basic 语句是(　　)。

(6) 可以生成输入对话框和消息对话框的函数分别是(　　)和(　　)。

(7) 如果强制用户声明变量时,需要在窗体的(　　)部分添加(　　)语句,或者通过 Visual Basic 6.0 的(　　)菜单的"选项"对话框来设置。

(8) Visual Basic 6.0 中,用来产生随机数的函数为(　　)。

(9) 布尔型变量的取值只有(　　)和(　　)。

(10) 语句 Dim a, b As Integer 中的 a 是(　　)类型变量。

3.选择题

(1) 设 a=5,b=4,c=3,d=2,表达式 3>2 * b Or a=c And b<>c Or c>d 的值是(　　)。

A. 1　　B. True　　C. False　　D. 2

【出处】2005年4月全国计算机等级考试二级 Visual Basic

(2) 设 a="MicrosoftVisualBasic",则以下使变量 b 的值为"VisualBasic"的语句是(　　)。

A. b=Left(a,10)　　B. b=Mid(a,10)　　C. b=Right(a,10)　　D. b=Mid(a,11,10)

【出处】2005年4月全国计算机等级考试二级 Visual Basic

(3) 假定有如下的窗体事件过程:

```
Private Sub Form_Click()
  a$ ="Microsoft Visual Basic"
  b$ =Right(a$ ,5)
  c$ =Mid(a$ ,1,9)
  MsgBox a$ , 34, b$ , c$ , 5
End Sub
```

程序运行后单击窗体,则在弹出的信息框中的标题栏中显示的信息是(　　)。

A. Microsoft Visual Basic　　B. Microsoft　　C. Basic　　D. 5

【出处】2005年4月全国计算机等级考试二级 Visual Basic

(4) 设 a=2,b=3,c=4,d=5,表达式 Not a<=c Or 4 * c=b^2 And b<>a+c 的值是(　　)。

A. -1　　B. 1　　C. True　　D. False

【出处】2005年9月全国计算机等级考试二级 Visual Basic

(5) 设 a=5,b=10,则执行 c=Int((b-a) * Rnd+a)+1 后,c 值的范围为(　　)。

A. 5~10　　B. 6~9　　C. 6~10　　D. 5~9

【出处】2005年9月全国计算机等级考试二级 Visual Basic

(6) 在窗体上画一个命令按钮,名称为 Command1,然后编写如下事件过程:

```
Private Sub Command1_Click()
  a$ ="software and hardware"
  b$ =Right(a$ ,8)
  c$ =Mid(a$ ,1,8)
  MsgBox a$ ,,b$ ,c$ ,1
```

```
End Sub
```

运行程序，单击命令按钮，则在弹出的信息框的标题栏中显示的是(　　)。

A. software and hardware　　B. software　　C. hardware　　D. 1

【出处】2005 年 9 月全国计算机等级考试二级 Visual Basic

(7) 设有如下语句：

```
Dim a,b As Integer
c="VisualBasic"
d=#7/20/2005#
```

以下关于这段代码的叙述中，错误的是(　　)。

A. a 被定义为 Integer 类型　　B. b 被定义为 Integer 类型

C. c 中的数据是字符串　　D. d 中的数据是日期类型

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(8) 以下能从字符串"VisualBasic"中直接取出子字符串"Basic"的函数是(　　)。

A. Left　　B. Mid　　C. String　　D. Instr

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(9) 在窗体上画一个命令按钮和一个文本框，其名称分别为 Command1 和 Text1，把文本框的 Text 属性设置为空白，然后编写如下事件过程：

```
Private Sub Command1_Click()
  a=InputBox("Enter an integer")
  b=InputBox("Enter an integer")
  Text1.Text=b+a
End Sub
```

程序运行后，单击命令按钮，如果在输入对话框中分别输入 8 和 10，则文本框中显示的内容是(　　)。

A. 108　　B. 18　　C. 810　　D. 出错

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(10) 在 Visual Basic 6.0 中，InputBox 函数的返回值的类型为(　　)。

A. 数值型　　B. 字符串　　C. 逻辑型　　D. 日期型

4. 编程题

在窗体上添加 1 个标签和 1 个命令按钮，命令按钮的 Caption 属性"求两个数之和"。程序运行后，单击"求两个数之和"按钮，会依次弹出两个对话框，分别要求输入第一个数和第二个数，按要求输入两个数后，这两个数的和被显示在标签上。

第 3 章　Visual Basic 程序设计结构

程序是命令的序列，以什么样的顺序来执行命令，是在编写程序的时候决定的，与程序的控制结构有关。各种编程语言都提供了若干基本的控制结构，每一种控制结构中的程序执行流向有所不同。可以把程序看作是由若干个基本控制结构组成的实体，每一个基本结构又包含一个或多个语句，能够控制局部范围内的程序流向，那么若干个控制结构组合起来就会促使程序的流程向着完成其功能的方向发展。Visual Basic 提供了三种基本结构：顺序结构、选择结构和循环结构。

3.1　顺序结构

顺序结构是一种最简单、最基本的程序控制结构。

3.1.1　顺序结构的概念与流程

顺序结构按照语句的先后排列顺序逐条执行，执行流程如图 3－1 所示。在图 3－1 中，计算机按照程序段出现的顺序从上到下依次执行，先执行程序段 A，接着执行下面相邻的程序段 B，程序段一般由一条或多条语句组成。顺序结构可以看成是系统默认的控制结构，不需要专门的语句来控制。

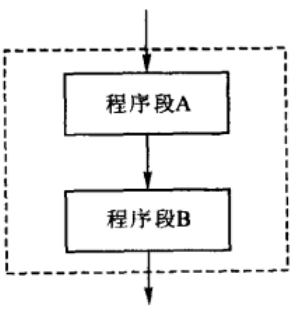


图 3－1　顺序结构

3.1.2　顺序结构的基本语句

1. 赋值语句

赋值语句是程序中最基本的语句，赋值语句可将指定的值赋给某个变量或对象的属性。赋值语句的一般格式为：

[Let] ＜变量＞＝＜表达式＞

或

[Let] [对象名.]＜属性名＞＝＜表达式＞

功能：将“＝”右边表达式的值，赋值给“＝”左边的变量或属性。这里的表达式可以是常量、字符串、变量(简单变量或下标变量)表达式及带有属性的对象。

例如：

```
Average = Sum/100                 '将表达式 Sum/100 的值赋给变量 Average
Text1.Text = Str(Average)         '把数值型变量 Average 转化为字符串赋给文本框
                                  Text1 的 Text 属性
```

说明：

(1) 语句定义符 Let 可以省略，通常书写程序时都将 Let 省略。

(2) 赋值语句具有计算和赋值的双重功能，当等号右边为表达式时，首先计算出表达式

的值,然后再赋值给等号左边的变量。

(3) 在向对象的属性赋值时,应指明对象名和属性名。系统默认的对象是当前窗口。

(4) 在 Visual Basic 中,"="既是赋值号也是逻辑等号,根据其所在语句中的位置不同,其作用有可能不同,一般 Visual Basic 程序是根据其所在的位置及前后文来判断"="是赋值号还是逻辑等号。

2.注释语句

为了提高程序的可读性,通常在 Visual Basic 应用程序的适当位置加上必要的注释,以对程序的作用加以说明。

注释语句的一般格式为:

'注释内容或 Rem 注释内容

功能:可对程序的有关内容起注释作用。

例如,程序中给出如下的注释语句:

'代码的功能为打印数据 a,b 的值

或 Rem 代码的功能为打印数据 a,b 的值

说明:

(1) 注释语句非执行语句,其对程序的执行过程不产生任何影响。它不被解释和编译,但在列程序清单时,注释内容将被完整地显示出来,以增加程序的可读性。

(2) 任何字符或汉字均可作为注释内容。

(3) 注释语句通常放在过程模块的开始部分作为程序的标题,也可放在程序行的末尾作为程序行注释,但注释语句不能放在续行符后面。

3.暂停语句

暂停语句主要用来把正在执行的解释程序设置为中断模式,以便用户对当前正在运行的程序进行检查和调试。

暂停语句的一般格式为:

Stop

功能:用来暂停程序的执行,同时打开立即窗口。它的作用类似于"运行"菜单中的"中断"命令。

通常用户在调试程序时,在程序中用 Stop 语句设置断点。当程序执行到该语句时,暂停程序的执行,但仍保持文件打开状态,并不退出 Visual Basic,这时用户可对程序进行相应的检查和调试。

但如果在可执行文件(.EXE)中含有 Stop 语句,当程序执行到该语句时,则将关闭所有文件。因此,一旦 Visual Basic 程序调试结束后,在生成可执行文件之前,应删去程序代码中所有的 Stop 语句。

4.结束语句

结束语句的一般格式为:

End

功能:结束一个程序的运行,可以放在过程中的任何位置以关闭代码执行,关闭以 Open 语句打开的文件并清除变量。返回操作系统(当程序编译成执行文件时)或 Visual Basic 系统(程序在 Visual Basic 集成环境解释执行时)。

说明：

(1) End语句只是生硬地终止代码的执行，它不调用Unload、QueryUnload或Terminate事件或任何其他的Visual Basic代码。

(2) Visual Basic中End语句的主要形式有End If、End Select、End Type、End With、End Sub、End Function等，它们与对应的语句配对使用。

(3) 在一个程序中有无End语句，对整个程序的执行不会产生影响，但如果程序中没有设置End语句或没有执行含有End语句的事件过程，则程序不能正常结束。

5.Exit语句

Exit语句用于退出Do…Loop、For…Next、Function或Sub代码块。对应的使用格式为Exit Do、Exit For、Exit Function或Exit Sub。

(1) Exit Do：提供一种退出Do…Loop循环的方法，并且只能在Do…Loop循环中使用。Exit Do会将控制权转移到Loop语句之后的语句。当Exit Do用在嵌套的Do…Loop循环中时，Exit Do会将控制权转移到Exit Do所在位置的外层循环。

(2) Exit For：提供一种退出For循环的方法，并且只能在For…Next或For Each…Next循环中使用。Exit For会将控制权转移到Next语句之后的语句。当Exit For用在嵌套的For循环中时，Exit For会将控制权转移到Exit For所在位置的外层循环。

(3) Exit Function：执行到该语句，程序立即从包含该语句的Function过程中退出。转回到调用Function过程的语句之后的语句继续执行。

(4) Exit Sub：执行到该语句，程序立即从包含该语句的Sub过程中退出。转回到调用Sub过程的语句之后的语句继续执行。

3.2 选择结构

在实际应用中，经常需要根据某一条件的成立与否来确定程序的执行流程，这种需要通过判断条件来决定程序执行流程的结构称为选择结构。

3.2.1 单行结构条件语句

单行结构条件语句格式为：

If 〈条件表达式〉 Then 〈语句块1〉 [Else〈语句块2〉]

功能：如果“〈条件表达式〉”为True，则执行“〈语句块1〉”；否则执行“〈语句块2〉”。

说明：

(1) 在上面的格式中，“〈条件表达式〉”是一个关系表达式或逻辑表达式。程序根据这个表达式的值(True或False)执行相应的操作。“〈语句块〉”是一个或多个Visual Basic语句，当含有多个语句时，各语句之间用冒号隔开。

(2) If语句中的“〈语句块2〉”是可选的，当省略该项时，If语句简化为

If 〈条件表达式〉 Then 〈语句块1〉

功能：如果“〈条件表达式〉”为True，则执行“〈语句块1〉”；否则执行下一行程序。

(3) 条件语句中的“〈语句块1〉”和“〈语句块2〉”都可以是条件语句，即条件语句可以嵌套。其深度(嵌套层数)没有具体规定，但受到每行字符数(1 024)的限制。

【例3-1】 输入3个数a,b,c，求其中的最大数。

解题思路:创建应用程序的用户界面和设置对象属性。在窗体上建立 4 个标签、4 个文本框和 1 个命令按钮,如图 3-2 所示。4 个文本框的 Name 属性从上到下依次为 Text1、Text2、Text3 和 Text4,它们的 Text 属性为空。命令按钮名称为 Command1,其 Caption 属性值为"判断"。

用户在"a="文本框(Text1)、"b="文本框(Text2)和"c="文本框(Text3)中输入数据,单击"判断"按钮后,在"最大数="文本框(Text4)中输出结果。程序如下:

```
Private Sub Command1_Click()
  Dim a As Integer, b As Integer
  Dim c As Integer, m As Integer
  a = Val(Text1.Text)
  b = Val(Text2.Text)
  c = Val(Text3.Text)
  If a > b Then m = a Else m = b
  'm 用来存放较大值
  If c > m Then m = c
  Text4.Text = m
End Sub
```

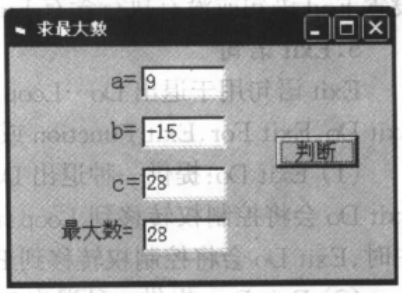


图 3-2 例 3-1 运行结果

3.2.2 IIf 函数

IIf 函数可用来执行简单的条件判断操作,它是"If…Then…Else"结构的简写版本,IIf 是"Immediate If"的缩略。

IIf 函数的格式为:

Result = IIf(〈条件表达式〉,〈语句块 1〉,〈语句块 2〉)

功能:"Result"是函数的返回值,"〈条件表达式〉"是一个关系表达式。当"〈条件表达式〉"为 True 时,IIf 函数返回"〈语句块 1〉",而当"〈条件表达式〉"为 False 时,IIf 函数返回"〈语句块 2〉"。"〈语句块 1〉"或"〈语句块 2〉"可以是表达式、变量或其他函数。

注意:IIf 函数中的 3 个参数都不能省略,而且要求"〈语句块 1〉"、"〈语句块 2〉"及结果变量的类型保持一致。

例如,将 a,b 中的较小者放入 Min 变量中,语句如下:

Min = IIf(a < b,a,b)

3.2.3 块结构条件语句

在条件比较简单并且分支语句块也很简单的情况下,为了书写方便可以用单行结构条件语句,但是这种条件语句不便进行程序块的凹凸处理,致使程序的结构不清晰,因此一个好的程序通常采用块结构的条件语句,这体现了良好的程序设计风格。

块结构条件语句格式:

```
If 〈条件表达式〉 Then
  〈语句块 1〉          '条件为真时,执行语句块 1
[ElseIf〈条件表达式 2〉Then
  〈语句块 2〉]         '条件为假时,执行语句块 2
[ElseIf〈条件表达式 3〉Then
```

```
    〈语句块 3〉]
  …
  [Else
    〈语句块 n〉]
  End If
```

功能：如果“〈条件表达式〉”为 True，则执行“〈语句块 1〉”；否则判断〈条件表达式 2〉…。

说明：

(1) 在块形式中，If 语句必须是第 1 行语句。If 块必须以 End If 语句结束。

(2) 当程序执行到 If 块时，首先测试“〈条件表达式〉”。如果“〈条件表达式〉”为 True，则执行 Then 之后的语句；如果条件为 False，则顺序执行。

(3) 这里的“〈语句块〉”可以是一个语句，也可以是多个语句。当有多个语句时，可以分别写在多行里，若写在一行中，则各语句之间用冒号隔开。

(4) Else 子句是可选的。

与单行条件语句相比，块结构条件语句有很多优点。例如，块形式比单行形式提供更好的结构和灵活性，它允许条件分支跨越数行。同时，用块形式可以测试更复杂的条件。块形式是程序的结构按逻辑引导，而不是把多个语句放在一行中。此外，使用块形式的程序一般容易阅读、维护和调试。任何单行形式的条件语句都可以改写成块形式。

【例 3－2】 使用块结构条件语句实现例3－1的功能。

程序代码如下，运行结果完全和例 3－1 相同。

```
Private Sub Command1_Click()
  Dim a As Integer, b As Integer
  Dim c As Integer, m As Integer
  a=Val(Text1.Text)
  b=Val(Text2.Text)
  c=Val(Text3.Text)
  If a>b Then
    m=a                    'm 用来存放较大值
  Else
    m=b
  End If
  If c>m Then
    m=c
  End If
  Text4.Text = m
End Sub
```

3.2.4　条件语句的嵌套

如果在条件语句的 Then 下的〈语句块 1〉中或在 Else 下的〈语句块 2〉中还含有条件语句，则称其为条件语句的嵌套。在程序设计中常用凹进的方法来表示嵌套的层次。

条件语句嵌套的一般格式如下：

```
If <条件 1> Then
   …
  If <条件 2> Then
   …
  Else
   …
  End If
   …
Else
   …
End If
```

说明:实际上,在所嵌套的条件语句中还可以含有条件语句,如果在所嵌套的条件语句中还可以含有条件语句,这就构成了条件语句的多层嵌套,但是嵌套层数是有限制的。

【例 3-3】 根据不同的时间段发出问候语,如 0 时~12 时,显示"早上好"。

解题思路:本例采用默认的用户界面,利用窗体装载(Load)事件,采用 Print 方法直接在窗体上输出结果。程序运行界面如图 3-3 所示,程序代码如下:

```
Private Sub Form_load()
  Dim h As Integer
  Show
  h=Hour(Time)                    '取系统的时间
  FontSize=30                     '设置字体
  ForeColor=RGB(255,0,0)          '设置前景颜色
  BackColor=RGB(255,255,0)        '设置背景颜色
  If h<12 Then
    Print "早上好!"
  Else
    If h<18 Then
      Print "下午好!"
    Else
      Print "晚上好!"
    End If
  End If
End Sub
```

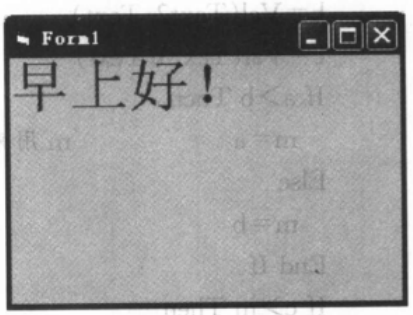


图 3-3 例 3-3 运行结果

【例 3-4】 使用块 If 格式实现例 3-3 的功能。

程序代码如下,运行结果完全和例 3-3 相同。

```
Private Sub Form_load()
  Dim h As Integer
  Show
  h=Hour(Time)                    '取系统的时间
```

```
    FontSize = 30                    '设置字体
    ForeColor = RGB(255,0,0)         '设置前景颜色
    BackColor = RGB(255,255,0)       '设置背景颜色
    If h<12 Then
      Print "早上好!"
    ElseIf h<18 Then
      Print "下午好!"
    Else
      Print "晚上好!"
    End If
  End Sub
```

3.2.5　多分支选择结构语句

虽然使用条件语句的嵌套可以实现多分支选择，但结构不够简明。Visual Basic 提供了另一种结构更清晰的多分支语句，即 Select Case…End Select 语句结构。多分支选择结构语句也可称为多情况选择语句，它比起上述条件语句嵌套更有效、更易读并且更易于跟踪测试。多情况选择语句一般格式为：

```
Select Case <测试表达式>
  [Case <表达式列表 1>
        <语句块 1>
  [Case <表达式列表 2>
        <语句块 2>]
  …
  [Case <表达式列表 n>
        <语句块 n>]
  [Case Else
        <语句块 n+1>]
End Select
```

功能：多情况选择语句以 Select Case 开头，以 End Select 结束。其执行时，根据“<测试表达式>”的值，从多个语句块中选择符合条件的一个语句块执行。

说明：

(1) 多情况选择语句中含有多个参数，这些参数的含义分别是：

① <测试表达式>：可以是数值表达式或字符串表达式，通常为变量或常量。

② <语句块 1>、<语句块 2>、……：每个语句块由一行或多行合法的 Visual Basic 语句组成。

③ <表达式列表 1>、<表达式列表 2>、…：称为域值，可以是下列形式之一：

〈表达式〉[,〈表达式〉]…。　　例如：Case 2,4,6,8

〈表达式〉To〈表达式〉。　　例如：Case 1 To 5

Is〈关系运算表达式〉，使用的运算符包括：<，<=，>，>=，<>，=。

例如：Case Is = 12，Case Is < a+b 等

"〈表达式列表〉"中的表达式必须与"〈测试表达式〉"的数据类型相同。

(2) 根据(1)得知,"〈表达式列表〉"上述 3 种形式,在具体使用时应注意以下几点:

① 关键字 To 用来指定一个范围。在这种情况下,必须把较小的值写在前面,较大的值写在后面,字符串常量的范围必须按字母顺序写出,例如:

```
Case  -5  To  -1
Case  "A"  To  "Z"
```

② 如果使用关键字 Is,则只能用关系运算符。例如:

```
Case Is<10
```

表示当测试表达式小于 10 时,执行相应的语句块。

注意:当用关键字 Is 定义条件时,只能是简单的条件,不能用逻辑运算符将两个或多个简单条件组合在一起。例如:Case Is>20 and Is<40 是不合法的。

③ 在一个 Select Case 语句中,3 种形式可以混用。例如:

```
Case Is<2,4,6,8,Is>20
```

表示当测试表达式的值小于 2 或者大于 20 或者是 4,6,8 时,执行相应的语句块。

(3) 多情况选择语句的执行过程是:先对"〈测试表达式〉"求值,然后测试该值与哪一个 Case 子句中的"〈表达式列表〉"相匹配。如果找到了,则执行与该 Case 子句有关的语句块,并把控制转移到 End Select 后面的语句;如果没有找到,则执行与 Case Else 子句有关的语句块,然后把控制转移到 End Select 后面的语句。

(4) 如果同一个域值的范围在多个 Case 子句中出现,则只执行符合条件的第一个 Case 子句的语句块。例如:

```
Select Case n
  Case 1,2
    Print "A"
  Case 2 To 5
    Print "B"
  Case Is>2
    Print "C"
End Select
```

当 n=5 时,满足第 2 个和第 3 个分支的条件,但只执行排在前面的第 2 个分支的语句,输出 B。

(5) 在多情况选择语句中,Case Else 必须放在所有的 Case 子句之后。如果在 Select Case 结构中的任何一个 Case 子句都没有与测试表达式相匹配的值,而且也没有 Case Else 子句,则不执行任何操作。

【例 3-5】 输入学生成绩(百分制),判断该成绩的等级(优良、及格、不及格)。

解题思路:创建应用程序的用户界面和设置对象属性。在窗体上建立 2 个标签、1 个文本框(Text1:用于输入学生的成绩,其 Text 属性为空)和 1 个命令按钮(Command1:其 Caption 属性为"执行")。

用户从"成绩"文本框中输入学生成绩,单击"执行"按钮,经判断得到成绩等级并显示在窗体上。程序的运行结果如图 3-4 所示。

程序代码如下：

```
Private Sub command1_click()
  Dim score As Integer,temp As String
  score=Val(Text1.Text)
  temp="成绩等级为:"
  Select Case score
  Case 0 To 59
    Label2.Caption=temp+"不及格"
  Case 60 To 79
    Label2.Caption=temp+"及格"
  Case 80 To 100
    Label2.Caption=temp+"优良"
  Case Else
    Label2.Caption="成绩出错"
  End Select
End Sub
```

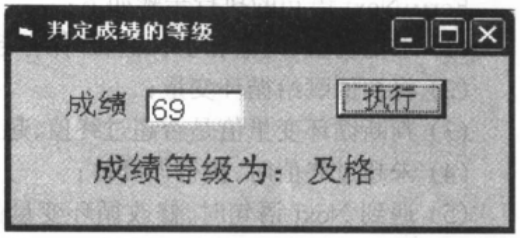


图 3-4　例 3-5 运行结果

3.3　循环结构

循环结构是一种重复执行的程序结构，其基本操作是在指定的条件下多次重复执行一组语句，被重复执行的这组语句称为循环体。通常循环结构有“当型循环”(先判断条件，后执行循环)和“直到型循环”(先执行循环，再判断条件)两种。在 Visual Basic 中，实现循环结构的语句主要有 4 种：For…Next 语句、Do While/Until…Loop 语句、Do…Loop While/Until 语句、While…Wend 语句。

3.3.1　For…Next 语句

For 语句一般用于循环次数已知的循环，其基本格式如下：

```
For 〈循环变量〉=〈初值〉 to 〈终值〉 [Step 〈步长〉]
    [语句块]
    [Exit For]    } 循环体
    [语句块]
Next〈循环变量〉
```

说明：

(1) 当“〈初值〉”<“〈终值〉”时，“〈步长〉”>0；如果省略，则系统默认步长为 1。

(2) 当“〈初值〉”>“〈终值〉”时，“〈步长〉”<0；如果步长为 0，而循环体中又无退出循环的语句(如 Exit For)，则将构成死循环。

(3) Exit For 用来结束 For 语句，通常使用时，总是出现在 If 语句或 Select Case 语句内部，内嵌在循环语句中。

(4) 循环次数：$N=\mathrm{Int}\left(\dfrac{\text{终值}-\text{初值}}{\text{步长}}+1\right)$

For…Next 语句的执行步骤如下：

(1) 求出初值、终值和步长值，并保存起来；

(2) 将初值赋给循环变量；

(3) 判断循环变量值是否超过终值，超过终值时，退出循环，执行 Next 之后的语句；

(4) 未超过终值时，执行循环体；

(5) 遇到 Next 语句时，修改循环变量值，即把循环变量的当前值加上步长值后再赋给循环变量；

(6) 转到(3)去判断循环条件，决定下一步是否继续执行循环。

【例 3-6】 求 Sum=1+2+3+…+10，把结果显示在窗体上。

解题思路：采用 Print 直接在窗体上输出结果，程序代码如下：

```
Private Sub Form_Load()
  Show
  Sum=0
  For k=1 To 10
    Sum=Sum+k
  Next k
  Print "Sum=";Sum
End Sub
```

根据 For…Next 语句的执行步骤，程序运行结果如下：

Sum=55

3.3.2 Do…Loop 语句

For…Next 语句主要用在已知循环次数的情况下，如果事先不知道循环次数，通常要使用 Do…Loop 语句。Do…Loop 语句有两种格式：当型循环结构和直到型循环结构。

1. 当型循环结构

```
Do {While|Until}<条件>
   [语句块]
   [Exit Do]    } 循环体
   [语句块]
Loop
```

2. 直到型循环结构

```
Do
   [语句块]
   [Exit Do]    } 循环体
   [语句块]
Loop{While|Until }<条件>
```

说明：

(1) 当使用“While <条件>”构成循环时，“<条件>”为“true”，则反复执行循环体；“<条件>”为“fales”，则退出循环。此时当型循环和直到型循环执行过程分别如图 3-5 和 3-6 所示。

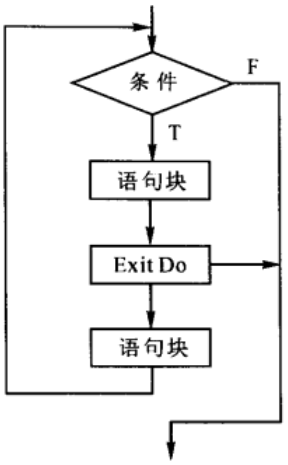


图 3－5　Do While…Loop 执行过程

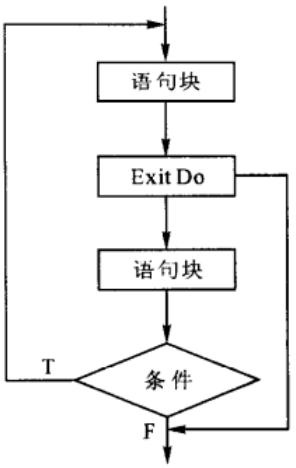


图 3－6　Do…Loop While 执行过程

(2) 当使用“Until ＜条件＞”构成循环时，“＜条件＞”为“False”，则反复执行循环体；“＜条件＞”为“True”，则退出循环。此时当型循环和直到型循环执行过程分别如图 3－7 和 3－8 所示。

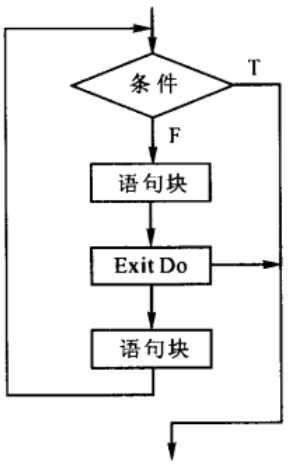


图 3－7　Do Until…Loop 执行过程

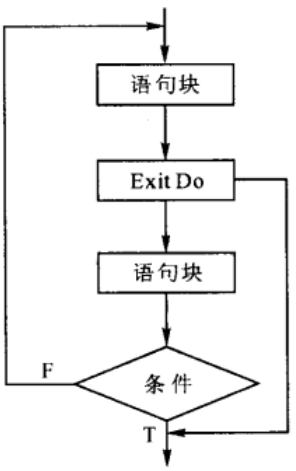


图 3－8　Do…Loop Until 执行过程

(3) 在循环体内一般应有一个专门用来改变条件表达式中变量的语句，以使随着循环的执行，条件趋于不成立(或成立)，最后达到退出循环的目的。

(4) 语句 Exit Do 的作用是退出它所在的循环结构，它只能用在 Do…Loop 结构中，并且常常是同选择结构一起出现在循环结构中，用来实现当满足某一条件时提前退出循环。

【例 3－7】 分别使用4 种形式的 Do…Loop 循环语句计算从 1 加到 100 的结果，把结果显示在窗体上。

解题思路：采用 Print 直接在窗体上输出结果。

(1) 使用 Do While…Loop 循环语句实现，程序代码如下：

```
Private Sub Form_Load()
  Dim i As Integer, Sum As Integer
```

```
    Show
    i=1
    Sum=0
    Do While i<=100
        Sum=Sum+i
        i=i+1
    Loop
    Print "Sum=";Sum
End Sub
```

程序运行结果:Sum=5050

(2) 使用 Do…Loop While 循环语句实现,运行结果相同,程序代码如下:

```
Private Sub Form_Load()
    Dim i As Integer, Sum As Integer
    Show
    i=1
    Sum=0
    Do
      Sum=Sum+i
      i=i+1
    Loop While i<=100
    Print "Sum=";Sum
End Sub
```

(3) 使用 Do Until…Loop 循环语句实现,运行结果相同,程序代码如下:

```
Private Sub Form_Load()
    Dim i,Sum As Integer
    Show
    i=1
    Sum=0
    Do Until i>100
        Sum=Sum+i
        i=i+1
    Loop
    Print "Sum=";Sum
End Sub
```

(4) 使用 Do…Loop Until 循环语句实现,运行结果相同,程序代码如下:

```
Private Sub Form_Load()
    Dim i,Sum As Integer
    Show
    i=1
```

```
    Sum = 0
    Do
      Sum = Sum + i
      i = i + 1
    Loop Until i > 100
    Print "Sum = "; Sum
End Sub
```

3.3.3　While…Wend 语句

While…Wend 语句基本格式为：

```
While <条件>
  <循环体>
Wend
```

功能：当“<条件>”为真时执行循环体；当“<条件>”为假时终止循环。

说明：

该语句的功能与上述 Do While…Loop 循环语句相似，二者的差别是：While…Wend 语句中不能使用循环出口语句 Exit 跳出循环。

【例 3-8】 输入两个正整数，求它们的最大公约数。

解题思路：

(1) 对于两个正整数 m，n，若求二者的最大公约数，首先令 m>n；

(2) 令 m 除以 n 得到余数 r；

(3) 若 r=0，则 n 为求得的最大公约数，程序结束；否则执行第(4)步；

(4) m←n，n←r，再重复执行第(2)步。

创建应用程序的用户界面和设置对象属性。在窗体上建立 4 个标签、3 个文本框和 1 个命令按钮。3 个文本框的名称分别为 Text1、Text2 和 Text3，Text 属性均为空。按钮名称为 Command1，其 Caption 属性为“计算”。当用户在“m=”文本框和“n=”文本框中输入数据后，单击“计算”按钮，则在“最大公约数”文本框中输出结果。

程序运行结果如图 3-9 所示，程序代码如下：

```
Private Sub command1_click()
  Dim m As Integer, n As Integer, r As Integer
  m = Val(Text1.Text)
  n = Val(Text2.Text)
  If m <= 0 Or n <= 0 Then
    MsgBox "数据错误!"
    End
  End If
  If m < n Then
    t = m
    m = n
    n = t
```

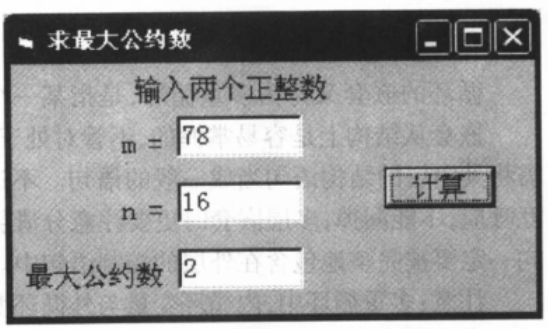


图 3-9　例 3-8 运行结果

```
    End If
    r = m Mod n
    While (r <> 0)
      m = n
      n = r
      r = m Mod n
    Wend
    Text3.Text = n
End Sub
```

3.3.4 Goto 语句

Goto 语句将指针无条件地转移到标号或行号指定的那行语句。其格式为：

Goto 〈标号|行号〉

其中：

〈标号〉是任何字符的组合，不区分大小写，必须以字母开头，以冒号结尾。标号必须是放在行的开始位置。

〈行号〉是任何数值的组合，在使用行号的模块内，该组合是惟一的。行号必须是放在行的开始位置。

说明：

(1) Goto 语句只能跳到它所在过程中的行。

(2) 在一个过程中，标号或行号都必须是惟一的。

(3) 利用 Goto 语句可以实现循环结构，此时往往还要和 If 语句相配合。基本形式为：

```
<语句标号>:If <条件表达式> Then
              <程序段>
              goto <语句标号>
           End If
```

(4) Goto 语句是非结构化语句，过多地使用 Goto 语句，会使程序代码不容易阅读及调试。建议尽可能地少用或者不用 Goto 语句，而使用结构化控制语句。

3.4 循环的嵌套

3.4.1 循环嵌套的概念

循环的嵌套又称为多重循环，是指某一个循环结构内又出现另一个或多个循环结构。

嵌套从结构上是容易掌握的，不管对处于哪一层的结构语句而言，使用规则仍然不变，只需将其内层的结构语句当成一般的语句。不过要注意语句的完整，每个结构语句的子句要相互呼应，不能漏掉；多层嵌套时更要注意分清层次，不能出现相互交叉的情况，即内层的控制语句一定要被完整地包含在外层的控制语句中。书写上最好采用分层递进的书写格式。

注意：多重循环中，内循环变量与外循环变量不能同名。

多重循环嵌套的层数可以根据需要而定，嵌套一层称为二重循环，嵌套二层称为三重循环，以此类推。多重循环执行的过程是：外层循环每执行一次，内层循环就要从头开始执行一轮。

下面列出几种常见的二重循环嵌套形式：

(1)
```
For I=…
  …
  For J=…
    …
  Next J
  …
Next I
```

(2)
```
For I=…
  …
  Do While/Until…
    …
  Loop
  …
Next I
```

(3)
```
Do While…
  …
  For J=…
    …
  Next J
  …
Loop
```

(4)
```
Do While/Until…
  …
  Do While/Until…
    …
  Loop
  …
Loop
```

3.4.2　循环嵌套的应用举例

【例3-9】　取1元、2元、5元的硬币共10枚，付给25元钱，有多少种不同的取法？

解题思路：设1元硬币为a枚，2元硬币为b枚，5元硬币为c枚，根据题意可列出方程组如下：

$$\begin{cases} a+b+c=10 \\ a+2b+5c=25 \end{cases}$$

利用二重循环，设外循环的循环变量为a，a从0～10；设内循环的循环变量为b，b从0～10；而c=10－a－b且c不能小于0。程序中，逐一判断得到的a，b，c是否满足a+2b+5c=25，满足时即为一组结果。

采用Print直接在窗体上输出结果，程序运行界面如图3-10所示，程序代码如下：

```
Private Sub Form_Load()
  Show
  CurrentX=100                    '确定开始显示的x坐标
  CurrentY=1100                   '确定开始显示的y坐标
  Print,"5元","2元","1元"
  n=0                             '记录解的组数
  For a=0 To 10
    For b=0 To 10
      c=10-b-a
      If a+2*b+5*c=25 And c>=0 Then
        n=n+1
        Print "(";n;")",c,b,a
      End If
  Next b, a
```

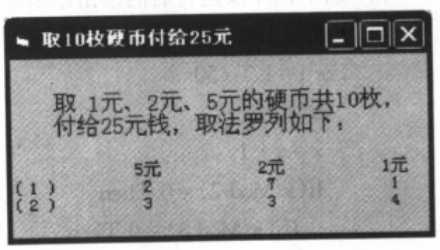


图3-10　例3-9运行结果

```
End Sub
```

习题三

1. 简答题

(1) Visual Basic 6.0 的基本程序设计结构分为哪几种？每种结构可以由什么语句实现？

(2) 在选择多分支结构语句时，什么情况下选用 Select Case 语句？有什么好处？

(3) 当型循环与直到型循环有什么不同？

(4) 比较 For 型循环语句与其他几种循环语句的异同。

(5) 为什么要引入 Exit 语句？什么情况下需要用到它？

(6) 什么是循环的嵌套？循环的嵌套对执行效率的影响如何？

2. 填空题

(1) 设 a=1、b=2、c=3、d=4，则表达式 IIF(a<b,a,IIF(c<d,a,d))的结果为(　　)。

(2) 以下程序段运行后的输出结果是(　　)。

```
n=9
Do While (n>6)
  n=n-1
Loop
Print n
```

(3) 下列程序段的执行结果为(　　)。

```
a=1
b=0
Select Case a
  Case 1
Select Case b
  Case 0
    Print "**0**"
  Case 1
    Print"**1**"
End Select
  Case 2
    Print"**2**"
End Select
```

(4) 以下程序段运行后的输出结果是(　　)。

```
Dim x,i
For i=1 To 50
  x=i
  x=x+1
  If(x Mod 2)=0 Then
    If (x Mod 3)=0 Then
      If (x Mod 7)=0 Then
        Print i
```

```
            End If
          End If
        End If
    Next i
```

(5) 有如下程序段,在运行时,输出 a 的值是(　　)。

```
    For i=1 To 3
      For j=5 To i Step-1
        For k=j To 6
          a=a+1
        Next k
      Next j
    Next i
    Print a
```

(6) 下列程序段的执行结果为(　　)。

```
    A=75
    If A>60 Then i=1
    If A>70 Then i=2
    If A>80 Then i=3
    If A>90 Then i=4
    Print "i=";i
```

(7) 执行以下程序段

```
    Dim x As Integer,i As Integer
    x=0
    For i=20 To 1 Step-2
      x=x+i·
    Next i
```

x 的值为(　　)。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic。

(8) 在窗体上画一个命令按钮,名称为 Command1,然后编写如下事件过程:

```
    Private Sub Command1_Click()
      Dim i As Integer, x As Integer
      For i=1 To 6
        If i=1 Then x=i
          If i<=4 Then
            x=x+1
          Else
            x=x+2
        End If
      Next i
      Print x
    End Sub
```

程序运行后,单击命令按钮,其输出结果为(　　)。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic。

3. 选择题

(1) Visual Basic 的基本控制结构是(　　)。

A. 顺序结构、选择结构和循环结构　　B. 单行结构、多行结构和块结构

C. 选择结构、多分支控制结构　　D. For 循环、"当型"循环结构、Do 循环结构

(2) 下面语句正确的是(　　)。

A. If X≥Y Then T=A:A=B:B=T　　B. If X≥Y Then T=A:A=B;B=T

C. If X>=Y Then T=A:A=B:B=T　　D. If X>=Y Then T=A: A=B: B=T

(3) 下面程序段的执行结果是(　　)。

```
i=1
s=0
while i<4
  i=i+1
  s=s+i^2
wend
print s
```

A. 10　　B. 13　　C. 29　　D. 54

(4) 下列循环执行的次数是(　　)。

```
a=0
For b=1 To-2 Step-1
  a=a-1
Next b
```

A. 2　　B. 4　　C. 3　　D. 0

(5) 设有如下程序:

```
Private Sub Command1_Click()
  Dim c As Integer,d As Integer
  c=4
  d=InputBox("请输入一个整数")
  Do While d>0
     If d>c Then
       c=c+1
     End If
     d=InputBox("请输入一个整数")
  Loop
  Print c+d
End Sub
```

程序运行后,单击命令按钮,如果在输入对话框中依次输入 1、2、3、4、5、6、7、8、9、0,则输出结果是(　　)。

A. 12　　B. 11　　C. 10　　D. 9

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic。

(6) 下列程序段的执行结果为(　　)。

```
x=5
y=-20
If Not x>0 Then x=y-3 Else y=x+3
```

```
Print x - y; y - x
```

A. -3　3　　B. 3　-3　　C. -3　-3　　D. 5　-8

(7) 在窗体上画一个命令按钮和一个文本框,名称分别为 Command1 和 Text1,然后编写如下程序:

```
Private Sub Command1_Click()
  a = InputBox("请输入日期(1~31)")
  t = "旅游景点:" & IIf(a>0 And a<=10,"长城","") & IIf(a>10 And a<=20,"故宫","") _
  & IIf(a>20 And a<=30,"颐和园","")
  Text1.Text = t
End Sub
```

程序运行后,如果从键盘输入 16,则在文本框中显示的内容是(　　)。

A. 旅游景点:长城故宫　　B. 旅游景点:长城颐和园

C. 旅游景点:颐和园　　D. 旅游景点:故宫

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(8) 下列程序段的执行结果为(　　)。

```
i = 4
a = 5
Do
  i = i + 1
  a = a + 2
Loop Until i >= 7
Print "i = "; i
Print "a = "; a
```

A. i = 4　a = 5　　B. i = 7　a = 13　　C. i = 8　a = 7　　D. i = 7　a = 11

(9) 下列程序段的执行结果为(　　)。

```
a = 0
b = 1
Do
  a = a + b
  b = b + 1
Loop While a < 10
Print a; b
```

A. 10　5　　B. a　b　　C. 0　1　　D. 10　30

(10) 下列程序段的执行结果为(　　)。

```
Sum = 0
For j = 1 To 10
  If j = 10 Then
  Else
    Sum = Sum + j
  End If
Next j
Print Sum
```

A. 45　　B. 55　　C. 显示错误信息　　D. 36

4. 编程题

(1) 设计一个对输入字符进行转换的程序。转换规则为:将其中的小写字母转换为大写字母,大写字母转换为小写字母,其余非字母字符均转换为“*”。在一个文本框中每输入一个字符,马上就进行判断和转换,转换后的结果显示在另一个文本框中。

(2) 编写程序:找出 100～999 之内满足下列条件的数据(水仙花数),各位数字的立方之和等于本数。例如:$153=1\times1\times1+5\times5\times5+3\times3\times3$。

(3) 编写程序:由键盘输入一个字符串,分别统计此字符串中数字字符、英文字母字符与其他字符的个数,把结果输出在 3 个不同的文本框中。

第 4 章　常用控件

Visual Basic 应用程序中的菜单、工具栏、按钮等组成部分都是具有属性、方法和事件的对象，把它们统称为控件。控件是 Visual Basic 的一种对象，是与用户交互的可视化部件，是 Visual Basic 程序设计的基础和重要元素，是设计应用程序界面的基本操作对象。控件放在窗体中以窗体为容器，它和窗体一样，具有属性、方法和事件，控件的属性和事件一般都是事先定义好的，当事件发生时，就会实现事件过程中程序代码要完成的功能。只有合理恰当地使用 Visual Basic 的各种控件，以及熟练掌握各种控件的属性、方法和事件，才能更好地开发 Visual Basic 的应用程序。

4.1　控件的分类

Visual Basic 的控件分为内部控件、ActiveX 控件和可插入对象三类。

1. 内部控件

内部控件也叫标准控件或常用控件，是由 Visual Basic 本身提供的在控件工具箱中默认出现的控件，不能从控件工具箱中删除，有 20 种，如图 4－1 所示。

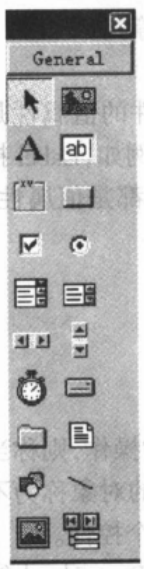


图 4－1　工具箱

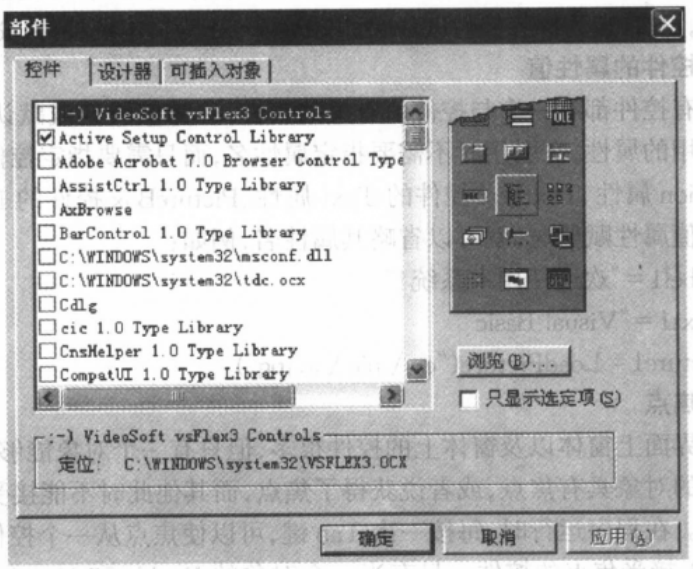


图 4－2　“部件”对话框

2. ActiveX 控件

ActiveX 控件是 Visual Basic 控件工具箱的扩展部分，这些控件在使用之前，必须先添加

到工具箱中。添加 ActiveX 控件的步骤如下：

(1) 用鼠标右击工具箱，出现快捷菜单。

(2) 选择快捷菜单中的“部件”命令，出现部件对话框，如图 4－2 所示。

(3) 单击“部件”对话框中的“部件”选项卡，然后单击复选框添加 ActiveX 控件。

(4) 单击“确定”按钮，则在工具箱中内部控件的下面出现添加的 ActiveX 控件。

3.可插入对象

可插入对象也叫外部控件或外部对象，是由其他应用程序创建的对象，利用可插入对象，就可以在 Visual Basic 应用程序中使用其他应用程序的对象。添加可插入对象到 Visual Basic 工具栏与添加 ActiveX 控件的方法相同，在“部件”对话框中，单击“可插入对象”选项卡，然后单击复选框添加需要的可插入对象，单击“确定”后，则在工具箱中内部控件的下面出现添加的可插入对象。

4.2 控件的通用特性

Visual Basic 的每个控件都有其不同的属性、方法和事件，但有些常用的属性、方法和事件是大部分控件都具有的。

1.名称(Name)属性

每个控件都具有名称(Name)属性，用于设置控件的名字。创建控件时，新对象的默认名称由对象类型名称后面加上一个惟一的整数组成。例如，第一个 CommandButton 控件的 Name 属性是 Command1，第二个是 Command2，第三个是 Command3，以此类推。

为了使控件的名称易记并体现其含义和代表性，建议按照以下的规则来命名，即名称前缀表示控件的类，其后为具有描述性含义的控件名称。例如，CmdOk 为命令按钮控件(CommandButton)，LblTitle 为标签控件(Lable)，TxtPassword 为文本框控件(TextBox)，等等。

2.控件的属性值

所有控件都有一个与控件值有关的属性，称为值属性或默认属性。控件的值属性是控件最常用的属性，在引用时不需要指定属性名，而只需要指定控件名即可。例如，Label 控件的 Caption 属性、TextBox 控件的 Text 属性、PictureBox 控件的 Picture 属性都是值属性，当给这些值属性赋值时，就可以省略其属性名，例如：

```
Label1 = "欢迎使用本系统!"
Text1 = "Visual Basic"
Picture1 = LoadPicture("c: \ pic \ a.jpg")
```

3.焦点

在界面上窗体以及窗体上的控件很多，但只有一个对象能够接受键盘的操作，则称当前被操作的对象具有焦点，或者说获得了焦点，而其他此时不能接受键盘输入的对象称为不具有焦点。在程序运行时，每按一次 Tab 键，可以使焦点从一个控件移到另一个控件。

(1) 接受焦点的控件。只有当一个对象的 Enable 和 Visible 属性均为 True 时，才能接受焦点；Frame、Label、Menu、Line、Shape、Image、Timer 控件都不能接受焦点。

(2) 将焦点赋予对象的方法。程序运行时，用鼠标或快捷键选择对象，或者按 Tab 键把焦点移到对象上；在程序代码中使用 SetFocus 方法。

对于可以接受焦点的大多数控件，从其外观上可以看出它是否具有焦点。例如，当命令按钮具有焦点时，其边框就会突出显示。

(3) 焦点事件。当对象获得焦点时，发生 GotFocus 事件；当对象失去焦点时，发生 LostFocus 事件。LostFocus 事件主要用来对控件的操作更新进行证实和有效性检查，或用于修改在对象的 GotFocus 事件过程中建立的条件。

(4) 焦点属性。TabIndex 属性决定它在 Tab 键中的顺序，按照建立对象的先后顺序的默认规定，从 0 开始；TabStop 属性用来指定焦点是否在对象上停留，默认属性值为 True，表示停留。

4. 访问键

菜单和命令按钮、复选框、选项按钮控件都可以具有访问键，也就是快捷键，通过键盘来访问控件。设置访问键的方法是在控件的 Caption 属性中把字符“&”加在访问字符的前面，当按“Alt + 访问字符”时就实现其功能。

5. 控件容器

Visual Basic 中的窗体(Form)、框架(Frame)、图片框(PictureBox)等控件都可以作为其他控件的容器，当移动容器时，容器中的控件也同时跟着移动。在容器中，控件的 Left、Top 属性值是指其在容器中的位置。

4.3　常用标准控件

在 Visual Basic 的控件工具箱中提供了 20 种标准控件，见表 4 - 1。下面介绍其中最常用的 12 种控件。

表 4 - 1　Visual Basic 6.0 标准控件

图标	控件名	类名	功能描述
	标签	Label	显示不可交互或不可修改的静态文本
	命令按钮	CommandButton	可完成命令或操作
	文本框	TextBox	可输入或显示文本
	框架	Frame	为控件提供可视化的功能化容器
	选项按钮	OptionButton	显示多个选项，用户只能从中选择一项。
	复选框	CheckBox	显示多个选项，用户可以选择多项
	列表框	ListBox	显示列表项目，用户可以从中选择
	组合框	ComboBox	文本框和列表框的组合，用户可输入或选择选项
	图像	Image	显示位图、图标或 Windows 图元文件、JPEG、GIF 文件
	图片框	PictureBox	显示位图、图标或 Windows 图元文件、JPEG、GIF 文件，可作容器使用
	水平滚动条	HScrollBar	为没有滚动条的控件添加水平滚动条
	垂直滚动条	VScrollBar	为没有滚动条的控件添加垂直滚动条

续表 4－1

图标	控件名	类名	功能描述
	时钟	Timer	按指定时间间隔执行定时器事件
	驱动器列表框	DriveListBox	显示有效的磁盘驱动器并允许用户从中选择
	目录列表框	DirListBox	显示目录和路径并允许用户从中选择
	文件列表框	FileListBox	显示文件列表并允许用户从中选择
	直线	Line	在窗体上添加线段
	形状	Shape	在窗体上添加不同的几何图形
	数据	Data	连接数据库，显示数据库中的数据
	OLE 容器	OLE	将其他应用程序嵌入到 Visual Basic 应用程序

4.3.1 标签(Label)控件

1.功能

标签(Label)控件用于显示不需要编辑的文本信息，一般用于在窗体上进行文字说明或信息提示。

2.常用属性

标签控件的常用属性见表 4－2。

表 4－2 标签控件的常用属性

属性	说　明
Caption	显示在标签中的文本，最多可有 1 024 个字符
AutoSize	是否能够自动改变大小以适应其中的内容。True 可自动适应，False 不能改变大小，超长的文本被截掉
BackStyle	背景样式，0 为透明，1 为不透明
Alignment	标签中文本的水平对齐方式，分为左、右、居中
WordWrap	设置标签是否根据文本长度进行保持宽度不变的垂直方向变化

3.方法

标签控件的常用方法有 Refresh 和 Move，Refresh 用于刷新内容。

读者根据标签控件的常用属性进行思考和实践。例如，用标签控件在窗体上显示“欢迎使用程序设计学习软件”，如图 4－3 所示。

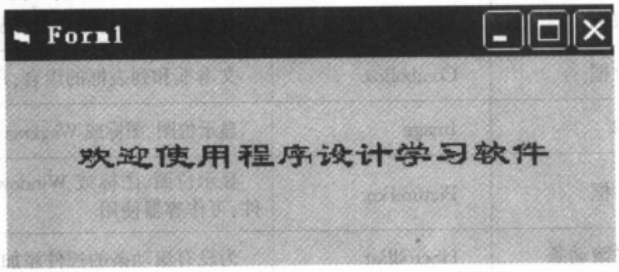


图 4－3 标签控件的应用

4.3.2　命令按钮(CommandButton)控件

1.功能

命令按钮(CommandButton)主要用作程序操作中的控制按钮，是 Windows 应用程序中最常用的控件，比如，程序运行操作中的“确定”、“取消”、“上一步”、“下一步”等。

2.属性

命令按钮最常用的属性见表 4－3。

表 4－3　命令按钮的常用属性

属　性	说　明
Caption	设置命令按钮上显示的文字
Enabled	设置按钮是否有效。True 有效，False 无效
Style	设置按钮外观。0 为标准按钮，1 为图形按钮
Default	设置是否为默认确定按钮。值为 True 时，按回车键即可
Cancel	设置是否为默认取消按钮。值为 True 时，选中该按钮，并执行其功能
ToolTipText	设置鼠标停留在按钮上时显示的提示文本
Picture	设置按钮上显示的图形。只有 Style 属性为 1 时，该属性生效
DownPicture	设置按下按钮时显示的图形。只有 Style 属性为 1 时，该属性生效
DisabledPicture	设置按钮无效时显示的图形。Style 属性为 1 时，该属性生效
Font	设置按钮上显示文本的字体、字形、字号

2.方法

命令按钮的常用方法是 SetFocus。

3.事件

命令按钮最常用的事件是 Click(单击)事件。以下情况可以发生 Click 事件：

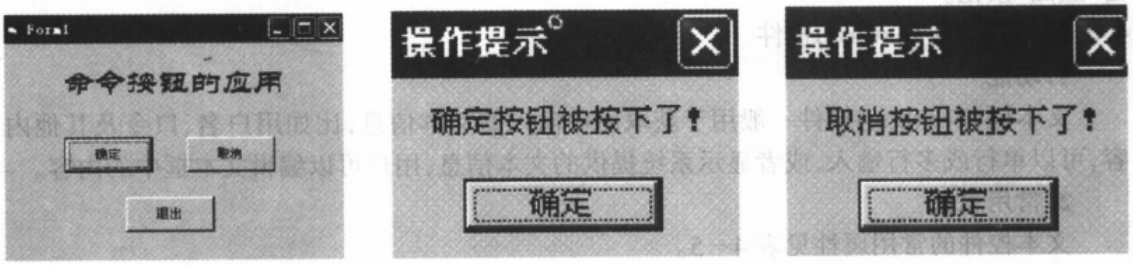
(1) 用鼠标单击按钮。

(2) 按钮具有焦点时，按回车键或空格键。

(3) 在程序代码中，将命令按钮的 Value 属性值设置为 True。

(4) 运行时通过按 Alt＋访问键。

【例 4－1】 编写程序体会命令按钮的属性和事件的应用。在窗体上添加 3 个命令按钮和 1个标签控件，在标签中显示“命令按钮的应用”，如图 4－4(a)所示。当单击“确定”或按下回车键时，弹出如图 4－4(b)所示的对话框，当单击“取消”或按下 Esc 键时，弹出如图 4－4(c)所示的对话框，当单击“退出”按钮时，将退出程序。观察程序运行结果并思考命令按钮的属性值设置。



(a)　　(b)　　(c)

图 4－4　例 4－1 程序运行结果

表 4-4 例 4-1 控件属性设置

对象	属性	属性值
Label1	Caption	命令按钮的应用
	Alignment	2 - Center
	Font	华文隶书二号字
	ForeColor	&H000000FF&
CmdOk	Caption	确定
	Font	宋体小五号字
	ForeColor	&H00C0C0C0&
	Default	True
CmdCancel	Caption	取消
	Font	宋体小五号字
	ForeColor	&H00C0C0C0&
	Cancel	True

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 界面设计。在窗体上添加 1 个标签和 3 个命令按钮控件，如图 4-4(a)所示。

(3) 属性值设置。修改标签和命令按钮的属性值，见表 4-4。

(4) 进入代码编辑窗口，编写如下事件过程：

```
Private Sub CmdOk_Click()
  MsgBox "确定按钮被按下了!", vbDefaultButton1, "操作提示"
End Sub
Private Sub CmdCancel_Click()
  MsgBox "取消按钮被按下了!", vbDefaultButton1, "操作提示"
End Sub
Private Sub CmdExit_Click()
  End
End Sub
```

(5) 运行程序，分别用鼠标单击“确定”、“取消”按钮，再分别按下回车键、Esc 键，最后单击“退出”按钮。

4.3.3 文本框(TextBox)控件

1. 功能

文本框(TextBox)控件一般用于获取用户输入的文本信息，比如用户名、口令及其他内容，可以单行或多行输入，或者显示系统提供的文本信息，用户可以编辑文本框中的内容。

2. 常用属性

文本控件的常用属性见表 4-5。

表 4-5 文本框的常用属性

属 性	说 明
Text	返回或设置文本框中的文本
PassWordChar	设置代替文本框中实际文本的字符
Locked	设置文本框是否可以编辑
MaxLength	设置文本框所能输入的最大字符数
MultiLine	设置文本框是否能够输入多行文本
SelText	设置/返回选中的文本字符串
SelStart	设置选中文本的起始位置
TabIndex	设置按下 Tab 键时文本框获得焦点的次序

3.方法

文本框的常用方法有 Refresh 和 SetFocus 等,其中 Refresh 方法用于刷新文本框中的内容。

4.事件

文本框可识别多种事件,如 GotFocus、LostFocus、Change、KeyPress、KeyDown、KeyUp、MouseMove、MouseUp 等,当文本框中的内容发生变化时,就会触发常用的 Change 事件。

4.3.4 框架(Frame)控件

框架(Frame)控件可以作为其他控件的容器,把窗体中的其他控件进行分组,因此也叫分组控件。框架控件的常用属性见表 4-6。

表 4-6 框架控件的常用属性

属 性	说 明
Caption	设置框架的标题。可用"&"符号添加热键,按下热键时,则组中第一个选中的控件获得焦点
Enable	可禁止用户使用框架中的所有控件
BorderStyle	设置 Caption 属性是否起作用

使用框架控件作为容器分组其他控件时,需要先在窗体中添加框架控件,然后再添加其它控件,这样移动框架时,其内部的所有控件才会随之移动。

4.3.5 选项按钮(OptionButton)控件

选项按钮(OPtionButton)也叫单选按钮,用于从一组互相排斥的选项按钮组中选择其中的一个按钮。如果用户选择了其中的一个按钮,则上一次被选中的按钮将自动清除被选中状态。选项按钮经常用于各种应用程序的界面和功能中,比如自测题、调查问卷、购物等。

选项按钮的常用属性见表 4-7,选项按钮的常用事件为 Click 事件。

表 4-7 选项按钮控件的常用属性

属 性	说 明
Caption	设置选项按钮的标题文本
Enabled	设置选项按钮是否有效
Alignment	设置选项按钮的按钮图标相对于其标题文本的对齐方式:0 为左对齐,1 为右对齐

续表 4-7

属性	说明
Picture	选项按钮上显示的图形
Value	设置选项按钮的状态：True 表示按钮被选中，则其他按钮该属性自动为 False
Style	设置选项按钮的样式：0 为标准样式，1 为图形样式
DownPicture	选项按钮被按下时显示的图形：当 Style 属性为 1 时有效
DisabledPicture	选项按钮被禁止使用时显示的图形：当 Style 属性为 1 时有效
Index	选项按钮作为控件数组元素时的索引值

【例 4-2】 利用选项按钮、标签按钮、文本框按钮设计一个程序，程序运行界面如图 4-5 所示，在文本框中，可以输入文字，然后单击选项按钮，文本框中的文字就会改变它在文本框中的对齐方式。

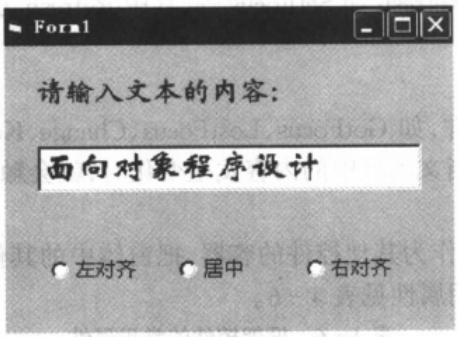


图 4-5 例 4-2 程序运行界面

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 程序界面设计。在窗体上添加 1 个标签、1 个文本框、3 个选项按钮控件，如图 4-5 所示。

(3) 属性值设置。各对象的属性设置见表 4-8。

表 4-8 例 4-2 控件属性设置

对象	属性	属性值
Label1	Caption	命令按钮的应用
	Alignment	2-Center
	Font	楷体四号字
	ForeColor	&H80000012&
Text1	Text	面向对象程序设计
	Font	华文新魏小二号字
	ForeColor	&H80000008&
Option1	Caption	左对齐
	Font	宋体五号字

续表 4-8

对象	属性	属性值
Option1	ForeColor	&H80000012&
	Alignment	0 - Left Justify
Option2	Caption	居中
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0 - Left Justify
Option3	Caption	右对齐
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0 - Left Justify

(4) 进入程序代码窗口,编写如下事件过程。

```
Private Sub Option1_Click()
  Text1.Alignment=0
End Sub
Private Sub Option2_Click()
  Text1.Alignment=2
End Sub
Private Sub Option3_Click()
  Text1.Alignment=1
End Sub
```

(5) 运行程序,运行结果如图 4-5 所示。单击选项按钮进行操作和思考。

4.3.6 复选框(CheckBox)控件

复选框(CheckBox)控件与选项按钮控件不同,可以从一组复选框中同时选中多个选项,而且可以在被选中和清除两个状态之间进行切换。在一组复选框中,每个复选框之间是独立的,用户可以选择一个或多个,可用于选择多个选项功能的程序操作。

复选框(CheckBox)控件的常用属性见表 4-9。

表 4-9　复选框控件的常用属性

属　性	说　明
Caption	设置选项按钮的标题文本
Enabled	设置选项按钮是否有效
Alignment	设置选项按钮的按钮图标相对于其标题文本的对齐方式:0 为左对齐,1 为右对齐
Picture	选项按钮上显示的图形
Value	设置选项按钮的状态:0 未选中,1 选中,2 禁止
Style	设置选项按钮的样式:0 为标准样式,1 为图形样式
DownPicture	选项按钮被按下时显示的图形。当 Style 属性为 1 时有效

续表 4-9

属性	说明
DisabledPicture	选项按钮被禁止使用时显示的图形。当 Style 属性为 1 时有效
Index	选项按钮作为控件数组元素时的索引值

复选框(CheckBox)控件的常用事件为 Click 事件。

【例 4-3】 利用复选框、选项按钮、框架、文本框、标签控件设计一个应用程序。程序运行界面如图 4-6 所示,当通过复选框、选项按钮进行字体和字形选择时,文本框中文本的字体和字号随着选择结果变化而变化。

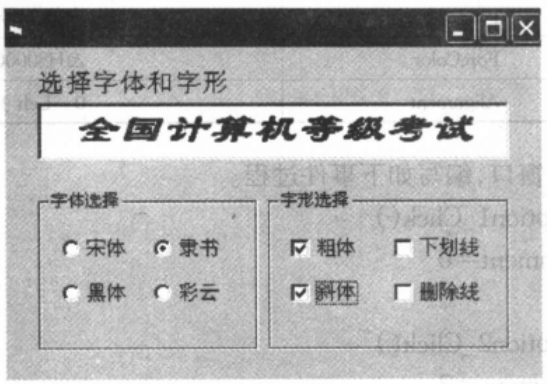


图 4-6 例 4-3 程序运行界面

程序设计步骤如下:

(1) 新建一个"标准 EXE"工程。

(2) 程序界面设计。在窗体上添加 1 个标签、1 个文本框、2 个框架、4 个复选框、4 个选项按钮控件,设计好其布局,如图 4-6 所示。

(3) 属性值设置。各对象的属性设置见表 4-10。

表 4-10 例 4-3 控件属性设置

对象	属性	属性值
Label1	Caption	选择字体和字形
	Alignment	2-Center
	Font	宋体四号字
	ForeColor	&H80000012&
Text1	Text	全国计算机等级考试
	Font	宋体二号字
	ForeColor	&H80000008&
Frame1	Caption	字体选择
	ForeColor	&H80000012&
	Enabled	True

续表 4－10

对象	属性	属性值
Frame2	Caption	字形选择
	ForeColor	&H80000012&
	Enable	True
Option1	Caption	宋体
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify
Option2	Caption	黑体
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify
Option3	Caption	隶书
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify
Option3	Caption	彩云
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify
Check1	Caption	粗体
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify
Check2	Caption	斜体
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify
Check3	Caption	下划线
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify
Check4	Caption	删除线
	Font	宋体五号字
	ForeColor	&H80000012&
	Alignment	0－Left Justify

(4) 进入程序代码窗口，编写如下事件过程：

```
Private Sub Option1_Click()
  Text1.FontName="宋体"
End Sub
Private Sub Option2_Click()
  Text1.FontName="黑体"
End Sub
Private Sub Option3_Click()
  Text1.FontName="隶书"
End Sub
Private Sub Option4_Click()
  Text1.FontName ="华文彩云"
End Sub
Private Sub Check1_Click()
  Text1.FontBold=Not Text1.FontBold
End Sub
Private Sub Check2_Click()
  Text1.FontItalic=Not Text1.FontItalic
End Sub
Private Sub Check3_Click()
  Text1.FontUnderline=Not Text1.FontUnderline
End Sub
Private Sub Check4_Click()
  Text1.FontStrikethrough=Not Text1.FontStrikethrough
End Sub
```

(5) 运行程序,单击复选框和选项按钮,对比这两个控件的区别。

4.3.7 列表框(ListBox)控件

列表框(ListBox)控件是 Windows 应用程序的常用控件,如图 4-7 所示。主要用于显示项目列表,列表中的项目是预先定义好的。列表中的用户可以从列表中选择其中的项目,可以选一个,也可以选多个。列表框显示列表方式可以是单列显示,也可以是多列显示,当列表中的选项数量超出其显示范围时,就会自动出现垂直滚动条。

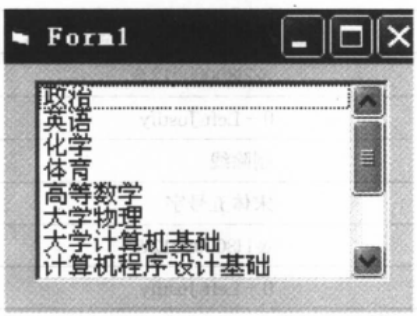


图 4-7 列表框

图 4-8 组合框

列表框(ListBox)控件的常用属性有：

(1) List 属性。List 属性用来列出或设置列表项的内容。其格式为：

字符串变量＝列表框名称. List(n)

列表框名称. List(n)＝"项目内容"

列表框名称. List(n)＝"要修改的项目内容"

例如：

Item2＝List1. List(1)

以上格式中的 n 为下标，即列表中项目的索引值，从 0 开始。

(2) ListIndex 属性。ListIndex 属性用来在程序运行时，设置或返回当前被选择项目的索引号。

(3) ListCount 属性。ListCount 属性用来在程序运行时，返回列表框中项目数量的数值。其格式为：

数值型变量＝列表框名称. ListCount

例如：

m＝List1. ListCount

(4) SelCount 属性。SelCount 属性用来返回列表框中所选项目数。

(5) Columns 属性。Column 属性用来设置或返回列表框的显示的列数。

(6) Selected 的属性。Selected 的属性用来设置或返回列表框中某个项目是否被选中。

(7) MultiSelect 属性。MultiSelect 属性用来在设计时，设置选项列表中的内容是否可以进行多重选择。

(8) Style 属性。Style 属性用来设置列表框的样式。

列表框(ListBox)控件常用的方法如下：

(1) AddItem 方法。用来为列表框添加新的项目。其格式为：

列表框名称. AddItem "项目"[,索引值]

例如：

List1. AddItem "计算机",10

(2) RemoveItem 方法。用来删除列表框中指定的项目。其格式为：

列表框名称. RemoveItem 索引值

例如：

List1. RemoveItem 5

(3) Clear 方法。用来清除列表空中的所有内容。其格式为：

列表框名称. Clear

列表框(ListBox)控件常用事件有 Click 事件和 DblClick 事件。

4.3.8　组合框(ComboBox)控件

组合框(ComboBox)控件是文本框(ListBox)和列表框(ListBox)的组合，它兼有文本框和列表框的功能，用户既可以从文本框中输入和修改文本，也可以从列表框中选择下拉的列表项。如图 4－8 所示。

列表框(ListBox)控件的大部分属性都可以作为组合框(ComboBox)控件的常用属性，如 Text、List、ListIndex、ListCount、Sorted 等。但是组合框也有自己的属性。

(1) Style 属性。用于设置组合框的类型和显示方式,共有 3 种类型。0 为下拉组合框;1 为简单组合框;2 为下拉列表框。

(2) Text 属性。用于返回用户选择的文本或直接在编辑区域输入的文本,可以在界面设置时直接输入。

组合框(ComboBox)控件的常用方法与列表框(ListBox)控件的常用方法相同,也有 AddItem 方法、RemoveItem 方法和 Clear 方法。

4.3.9 滚动条(ScrollBar)控件

滚动条(ScrollBar)控件通常用来附加到窗体上,它经常被用于移动页面已有内容,以便看到超出视窗区域的内容。滚动条有水平滚动条和垂直滚动条 2 种。

滚动条(ScrollBar)控件的常用属性如下:

(1) Max 属性。用来设置滚动条所能代表的最大值,范围在 -32 768~32 767 之间,缺省值为 32 767。

(2) Min 属性。用来设置滚动条所能代表的最小值,范围在 -32 768~32 767 之间,缺省值为 0。

(3) Value 属性。该属性表示滚动条内滑块所在位置所代表的值,其取值范围在 Max 和 Min 属性值之间。

(4) SmallChange 属性。单击滚动条两端箭头时,滑块移动的最小幅度,即 Value 值的最小增减量,其缺省值为 1。

(5) LargeChange 属性。单击滚动条两端箭头时,滑块移动的最大幅度,即 Value 值的最大增减量。

滚动条(ScrollBar)控件的常用事件有 Scroll 和 Change。

【例 4-4】 利用水平滚动条、垂直滚动条、文本框、标签控件编写一个程序,程序运行界面如图 4-9 所示。当单击滚动条时,代表滚动条滑块位置的数值就会显示在对应的文本框框中。

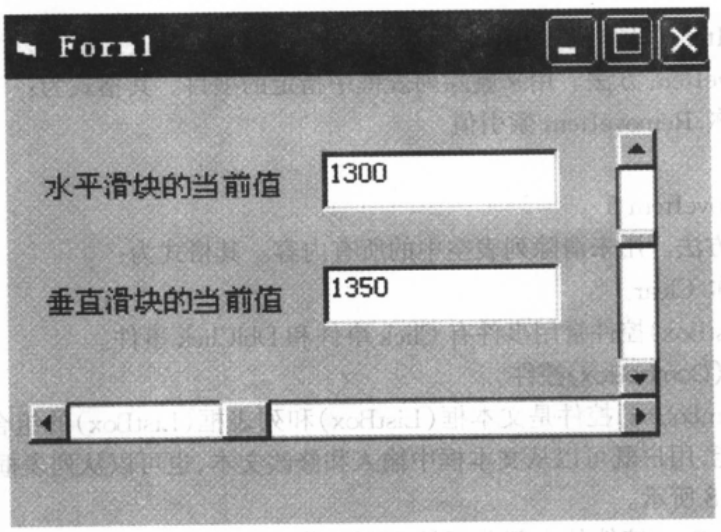


图 4-9 例 4-4 程序运行界面

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 程序界面设计。在窗体上添加 2 个标签、2 个文本框、1 个水平滚动条、1 个垂直滚动条控件，设计好其布局，如图 4－9 所示。

(3) 属性值设置。各对象的属性设置见表 4－11。

表 4－11　例 4－4 控件属性设置

对　象	属　性	属 性 值
Label1	Caption	水平滑块的当前值
	Alignment	2－Center
	Font	宋体小五号字
	ForeColor	&H80000012&
Label2	Caption	垂直滑块的当前值
	Alignment	2－Center
	Font	宋体小五号字
	ForeColor	&H80000012&
Text1	Text	""
	Font	宋体小五号字
	ForeColor	&H80000008&
Text2	Text	""
	Font	宋体小五号字
	ForeColor	&H80000008&
HScroll1	Max	2000
	Min	1000
	LargeChange	100
	SmallChange	100
VScroll1	Max	2000
	Min	1000
	LargeChange	50
	SmallChange	50

(4) 进入程序代码窗口，编写如下事件过程：

```
Private Sub HScroll1_Change()
  Text1.Text = CStr(HScroll1.Value)
End Sub
Private Sub VScroll1_Change()
  Text2.Text = CStr(VScroll1.Value)
End Sub
```

(5) 运行程序，单击滚动条上的滑块，观察文本框中数值的变化。

4.3.10 时钟(Timer)控件

时钟(Timer)控件又称计时器(Timer)控件,是 Visual Basic 系统提供给用户的一个计时器。由系统时钟控制,它能有规律地以一定时间间隔触发计时器事件(Timer)。计时器控件在设计时为一个时钟图标,而在运行时不可见,但是后台每隔一定时间,系统就会自动执行一次计时器事件。主要用于在程序中监控和控制事件过程,是一个非常有用的控件。

时钟(Timer)控件的常用属性有:

(1) Interval 属性。用来设置或返回计时事件间隔的毫秒数,取值范围为[0,65535]之间,间隔精确度不会超过 1/18 s。

(2) Enabled 属性。用来设置是否触发时钟事件。

时钟控件只有一个 Timer 事件,当设置的时间间隔到达时触发。如果 Interval 属性值不为 0 并且 Enabled 属性不为 False,则 Timer 事件就会不停地按照设置的时间间隔发生下去。

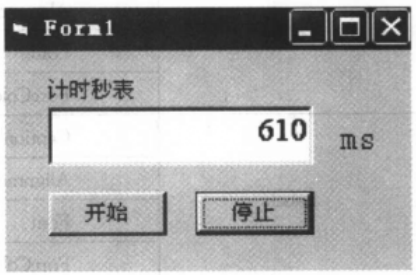


图 4-10 例 4-5 运行界面

【例 4-5】 编写一个"计时秒表"程序,程序运行界面如图 4-10 所示。

程序设计步骤如下:

(1) 新建一个"标准 EXE"工程。

(2) 程序界面设计。在窗体上添加 1 个计时器、2 个标签、1 个文本框、2 个命令控件,设计好其布局(图 4-10)。

(3) 属性值设置。各对象的属性设置见表 4-12。

表 4-12 例 4-5 控件属性设置

对象	属性	属性值
Label1	Caption	计时秒表
	Alignment	2 - Center Justify
	Font	宋体小五号字
	ForeColor	&H80000012&
Label2	Caption	ms
	Alignment	1 - Right Justify
	Font	Times New Roman 体四号字
	ForeColor	&H80000012&
Text1	Text	
	Font	宋体四号字
	ForeColor	&H80000008&
Command1	Caption	开始
Command2	Caption	停止
Timer1	Interval	10

(4) 进入程序代码窗口,编写如下事件过程:

```
Dim t As Integer
```

```
Private Sub Form_Load()
    Timer1.Interval = 10
    Timer1.Enabled = False
    Text1.Text = " 0 "
End Sub
Private Sub Timer1_Timer()
    t = t + 1
    If t > 1000 Then t = 0
    Text1.Text = CStr(t) + " 0 "
End Sub
Private Sub Command1_Click()
    Timer1.Enabled = True
End Sub
Private Sub Command2_Click()
    Timer1.Enabled = False
End Sub
```

(5) 运行程序,单击“开始”和“停止”按钮,观察文本框中的数字变化。

4.3.11　图像(Image)控件

图像(Image)控件是用来显示图像的控件,包括 Windows 位图、JPEG 图像、GIF 图像文件等。如图 4－11 所示。

图 4－11　图像控件

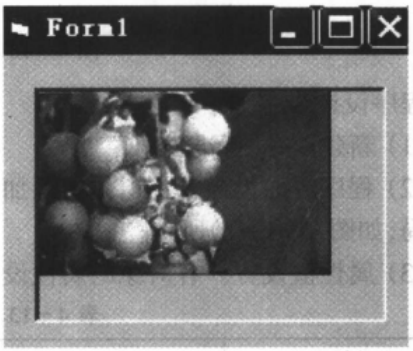


图 4－12　图片框控件

图像(Image)控件的常用属性有:

(1) Picture 属性。用于设置图像控件显示的图像所在的路径和文件名。

(2) Stretch 属性。用于设置图像框如何与图像进行大小相适应,True 为图像将自动适应图像框的大小,False 为图像框自动适应图像的大小。

4.3.12　图片框(PictureBox)控件

图片框(PictureBox)控件是用来显示图片的控件。包括 Windows 位图、JPEG 图像、GIF 图像、图表文件、文本等,同时也可以作为容器控件使用。如图 4－12 所示。

图片框(PictureBox)控件的常用属性有:

(1) Picture 属性。用于设置图片控件显示的图片所在的路径和文件名。

(2) AutoSize 属性。用于设置图片框如何与图片进行大小相适应，True 为图片框将自动调整其大小以适应图片的大小，False 为图片框的大小保持不变，当图片超出图片框的部分将被截掉。

4.4 应用实例

【例 4-6】 利用标签控件、文本框控件、命令按钮控件设计一个用户登录界面程序，如图 4-13(a)所示。如果用户分别或同时不输入用户姓名和用户密码而单击“确定”按钮，或者用户名和用户密码分别或者同时错误，程序都会弹出错误提示信息；只有用户名和密码同时正确时，程序才会提示“成功登陆!”，如图 4-13(b)所示(假设用户名为“student”，密码为“12345”)。

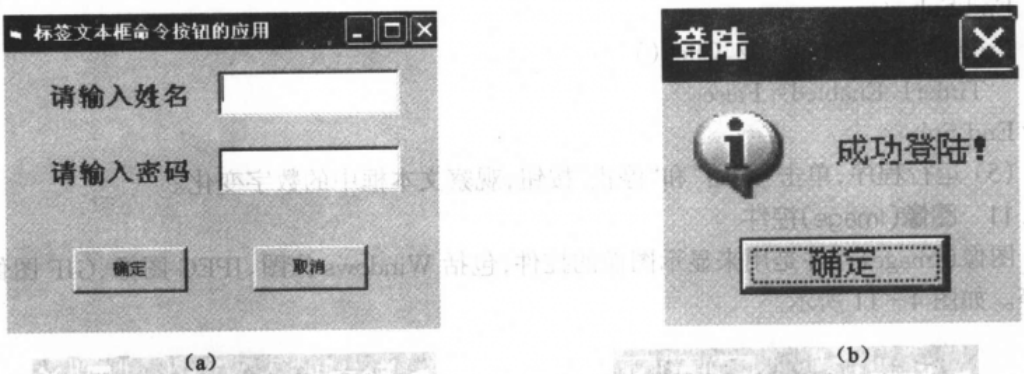


图 4-13 例 4-6 程序运行结果

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 程序界面设计。在窗体上添加 2 个标签、2 个文本框、2 个命令按钮控件，设计好界面布局，如图 4-13(a)所示。

(3) 属性值设置。各对象的属性设置见表 4-13。

表 4-13 例 4-6 控件属性设置

对象	属性	属性值
Label1	Caption	请输入姓名
	Alignment	2 - Center
	Font	宋体四号字加粗
	ForeColor	&H80000012&
Label2	Caption	请输入密码
	Alignment	2 - Center
	Font	宋体四号字加粗
	ForeColor	&H80000012&
Text1	Text	

续表 4－13

对象	属性	属性值
Text1	Font	宋体四号字
	ForeColor	&H80000008&
	PasswordChar	该属性不需设置
Text2	Text	
	Font	宋体四号字
	ForeColor	&H80000008&
	PasswordChar	该属性不需设置
CmdOk	Caption	确定
	Font	宋体小五号字
	ForeColor	&H00C0C0C0&
	Default	True
CmdCancel	Caption	取消
	Font	宋体小五号字
	ForeColor	&H00C0C0C0&
	Cancel	True

(4) 进入代码窗口中,编写如下事件过程:

```
Private Sub CmdOk_Click()
  If Text1.Text="" And Text2.Text="" Then
    MsgBox "用户名和密码不能为空,请重试!",vbInformation,"提示"
    Text1.Text=""
    Text2.Text=""
    Text1.SetFocus
  ElseIf Text1.Text="" Then
    MsgBox "用户名不能为空,请重试!", vbInformation,"提示"
    Text1.Text=""
    Text2.Text=""
    Text1.SetFocus
  ElseIf Text2.Text ="" Then
    MsgBox "密码不能为空,请重试!", vbInformation,"提示"
    Text1.Text=""
    Text2.Text=""
    Text1.SetFocus
  ElseIf Text1.Text <>"student" And Text2.Text<>"12345" Then
    MsgBox "用户名和密码不正确,请重试!", vbInformation,"提示"
    Text1.Text=""
    Text2.Text=""
```

```
        Text1.SetFocus
    ElseIf Text1.Text<>"student" Then
        MsgBox "用户名不正确,请重试!", vbInformation,"提示"
        Text1.Text=""
        Text2.Text=""
        Text1.SetFocus
    ElseIf Text2.Text<>"12345" Then
        MsgBox "密码不正确,请重试!",vbInformation,"提示"
        Text1.Text=""
        Text2.Text=""
        Text1.SetFocus
    Else
        MsgBox "成功登陆!",vbInformation,"登陆"
    End If
        Text1.Text=""
        Text2.Text=""
        Text1.SetFocus
End Sub
Private Sub CmdCancel_Click()
    Unload Me
End Sub
```

(5) 运行程序,分别或同时不输入用户名和用户密码、或者分别或同时输入错误的用户名和密码,观察程序的提示信息。

【例 4-7】 利用列表框、文本框、命令按钮、标签控件设计一个"显示和编辑省会、区号"的应用程序界面,如图 4-14 所示。

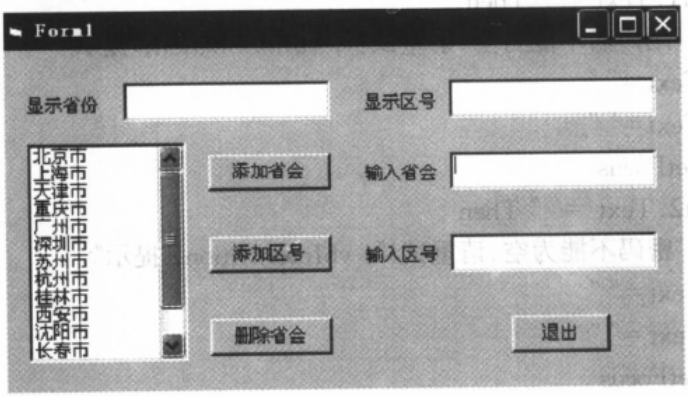


图 4-14 例 4-7 程序运行界面

程序设计步骤如下:

(1) 新建一个"标准 EXE"工程。

(2) 程序界面设计。在窗体上添加4个标签、4个文本框、4个命令按钮、1个列表框按钮控件，设计好界面布局，如图4－14所示。

(3) 属性值设置。各对象的属性设置见表4－14。

表4－14　例4－7控件属性设置

对象	属性	属性值
Label1	Caption	输入省会
Label2	Caption	输入区号
Label3	Caption	显示省会
Label4	Caption	显示区号
TxtInputP	Text	
TxtInputC	Text	
TxtShowP	Text	
TxtShowC	Text	
CmaAddP	Caption	添加省会
CmdAddC	Caption	添加区号
CmdDeltedP	Caption	删除区号
CmdExit	Caption	退出
List1	List	省会名

(4) 进入程序代码窗口，编写如下事件过程：

```
Private Sub CmdAddP_Click()
  List1.AddItem TxtInputP.Text, List1.ListCount
  TxtInputP.Text=""
End Sub
Private Sub CmdAddC_Click()
  List1.ItemData(List1.ListCount-1)=Val(txtInputC)
End Sub
Private Sub CmdDeletedP_Click()
  List1.RemoveItem List1.ListIndex
End Sub
Private Sub List1_Click()
  TxtShowP.Text=List1.Text
  txtShowC.Text="0"& List1.ItemData(List1.ListIndex)
End Sub
Private Sub CMdExit_Click()
  End
End Sub
Private Sub Form_Load()
  List1.AddItem "北京市"
```

```
    List1.AddItem "上海市"
    List1.AddItem "天津市"
    List1.AddItem "重庆市"
    List1.AddItem "广州市"
    List1.AddItem "深圳市"
    List1.AddItem "苏州市"
    List1.AddItem "杭州市"
    List1.AddItem "桂林市"
    List1.AddItem "西安市"
    List1.AddItem "沈阳市"
    List1.AddItem "长春市"
    List1.AddItem "哈尔滨市"
    List1.AddItem "石家庄市"
End Sub
```

(5) 运行程序,体会和思考它的功能。

习题四

1. 简答题

(1) Visual Basic 6.0 控件分为哪几类?

(2) Visual Basic 6.0 控件的通用特性有哪些?

(3) Visual Basic 6.0 的常用标准控件有哪些?

(4) 选项按钮与复选框有什么区别?

(5) 列表框与组合框有什么区别?

2. 填空题

(1) 窗体上有 1 个名称为 List1 的列表框,1 个名称为 Text1 的文本框,1 个名称为 Label1、Caption 属性为"Sum"的标签,1 个名称为 Command1、标题为"计算"的命令按钮。程序运行后,将把 1～100 之间能够被 7 整除的数添加到列表框中。如果单击"计算"按钮,则对 List1 中的数进行累加求和,并在文本框中显示计算结果,如图 4－15 所示。以下是实现上述功能的程序,请填空。

```
Private Sub Form_Load()
  For i=1 To 100
    If i Mod 7=0 Then
    (        )
    End If
  Next i
End Sub
Private Sub Command1_Click()
  Sum=0
  For i=0 To (      )
    Sum=Sum+(      )
  Next i
```

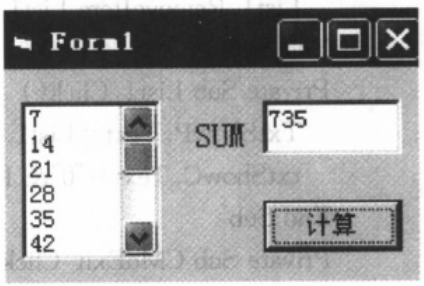


图 4－15 运行结果

```
    Text1.Text = Sum
  End Sub
```

【出处】2004年9月全国计算机等级考试二级 Visual Basic

(2) 在窗体上画1个文本框和1个图片框,然后编写如下两个事件过程:

```
  Private Sub Form_Click()
    Text1.Text = "VB程序设计"
  End Sub
  Private Sub Text1_Change()
    Picture1.Print "VBProgramming"
  End Sub
```

程序运行后,单击窗体,在文本框中显示的内容是(　　),而在图片框中显示的内容是(　　)。

【出处】2005年4月计算机等级考试二级 Visual Basic

3.选择题

(1) 在窗体上画1个文本框,然后编写如下事件过程:

```
  Private Sub Form_Click()
    x = InputBox("请输入一个整数")
    Print x + Text1.Text
  End Sub
```

程序运行时,在文本框中输入456,然后单击窗体,在输入对话框中输入123,单击"确定"按钮后,在窗体上显示的内容是(　　)。

A. 123　　B. 456　　C. 579　　D. 123456

【出处】2005年4月计算机等级考试二级 Visual Basic

(2) 在窗体上画1个文本框和1个计时器控件,名称分别为 Text1 和 Timer1,在属性窗口中把计时器的 Interval 属性设置为1000,Enabled 属性设置为 False,程序运行后,如果单击命令按钮,则每隔1s在文本框中显示一次当前的时间。以下是实现上述操作的程序:

```
  Private Sub Command1_Click()
    Timer1.(　　)
  End Sub
  Private Sub Timer1_Timer()
    Text1.Text = Time
  End Sub
```

在(　　)处应填入的内容是(　　)。

A. Enabled = True　　B. Enabled = False　　C. Visible = True　　D. Visible = False

【出处】2005年4月计算机等级考试二级 Visual Basic

(3) 假定在图片框 Picture1 中装入了一个图形,为了清除该图形(不删除图片框),应采用的正确方法是(　　)。

A. 选择图片框,然后按 Del 键

B. 执行语句 Picture1.Picture = LoadPicture("")

C. 执行语句 Picture1.Picture = ""

D. 选择图片框,在属性窗口中选择 Picture 属性,然后按回车键。

【出处】2005年4月计算机等级考试二级 Visual Basic

(4) 在窗体上画1个 List1 的列表框,1个名称为 Label1 的标签,列表框中显示若干个项目,当单击列表框中的某个项目时,在标签中显示被选中的项目的名称,下列能正确实现上述操作的程序是(　　)。

A. Private Sub List1_Click()
    Label1. Caption = List1. ListIndex
  End Sub

B. Private Sub List1_Click()
    Label1. Name = List1. ListIndex
  End Sub

C. Private Sub List1_Click()
    Label1. Name = List1. Text
  End Sub

D. Private Sub List1_Click()
    Label1. Caption = List1. Text
  End Sub

【出处】2005 年 4 月计算机等级考试二级 Visual Basic

(5) 在窗体上画一个命令按钮,名称为 Command1,然后编写如下事件过程:

```
Private Sub Command1_Click()
  Dim i As Integer,x As Integer
  For i=1 To 6
    If i=1 Then x=i
    If i <=4 Then
      x=x+1
    Else
      x=x+2
    End If
  Next i
  Print x
End Sub
```

程序运行后,单击命令按钮,其输出结果为(　　)。

A. 9　　B. 6　　C. 12　　D. 15

【出处】2005 年 4 月计算机等级考试二级 Visual Basic

(6) 在窗体上画一个名称为 Command1 的命令按钮,然后编写如下事件过程:

```
Private Sub Command1_Click()
  c="ABCD"
  For n=1 To 4
    Print (　　)
  Next
End Sub
```

程序运行后,单击命令按钮,要求在窗体上显示如下内容(　　):

```
D
CD
BCD
ABCD
```

则在(　　)处应填入的内容为(　　)。

A. Left(c,n)　　B. Right(c,n)　　C. Mid(c,n,1)　　D. Mid(c,n,n)

【出处】2005 年 4 月计算机等级考试二级 Visual Basic

(7) 在窗体(名称为 Form1)上画 1 个名称为 Text1 的文本框和 1 个名称为 Command1 的命令按钮,然后编写一个事件过程。程序运行以后,如果在文本框中输入一个字符,则把命令按钮的标题设置为"计算机等级考试"。以下能实现上述操作的事件过程是(　　)。

A. Private Sub Text1_Change()
    Command1. Caption = "计算机等级考试"

B. Private Sub Command1_Click()
    Caption = "计算机等级考试"

```
    End Sub
```

C.
```
Private Sub Form1_Click()
    Text1.Caption="计算机等级考试"
End Sub
```

D.
```
Private Sub Command1_Click()
    Text1.Text="计算机等级考试"
End Sub
```

【出处】2005年4月计算机等级考试二级 Visual Basic

(8) 设窗体上有1个文本框,名称为text1,程序运行后,要求该文本框不能接受键盘输入,但能输出信息,以下属性设置正确的是(　　)。

A. Text1.Maxlength=0　　B. Text1.Enabled=False

C. Text1.Visible=False　　D. Text1.Width=0

【出处】2004年9月计算机等级考试二级 Visual Basic

(9) 在窗体上画2个文本框,其名称分别为Text1和Text2,然后编写如下程序:

```
Private Sub Form_Load()
    Show
    Text1.Text=""
    Text2.Text=""
    Text1.SetFocus
End Sub
Private Sub Text1_Change()
    Text2.Text=Mid(Text1.Text,8)
End Sub
```

程序运行后,如果在文本框Text1中输入BeijingChina,则在文本框Text2显示的内容是(　　)。

A. BeijingChina　　B. China　　C. Beijing　　D. BeijingC

【出处】2006年4月全国计算机等级考试二级 Visual Basic

(10) 在窗体上画1个列表框和1个命令按钮,其名称分别为List1和Command1,然后编写如下事件过程:

```
Private Sub Form_Load()
    List1.AddItem "Item1"
    List1.AddItem "Item2"
    List1.AddItem "Item3"
End Sub
Private Sub Command1_Click()
    List1.List(List1.ListCount)="AAAA"
End sub
```

程序运行后,单击命令按钮,其结果为(　　)。

A. 把字符串"AAAA"添加到列表框中,但位置不能确定

B. 把字符串"AAAA"添加到列表框的最后(即"Item3"的后面)

C. 把列表框中原有的最后一项改为"AAAA"

D. 把字符串"AAAA"插入到列表框的最前面(即"Item1"的前面)

【出处】2006年4月全国计算机等级考试二级 Visual Basic

4. 编程题

利用标签、命令按钮、列表框编写一个程序,实现移动列表框中的项目的功能。

第 5 章　数组及应用

在实际应用中往往需要处理同一性质的成批数据。例如，要求 50 个学生的平均成绩，并统计高于平均成绩的学生人数。如果使用简单变量进行处理，就需要使用 a1，a2，…，a50，变量来存放，要输入 50 个学生的成绩，也就需要使用 50 个输入语句；要求平均值和统计大于平均值的学生人数，则程序代码繁多。处理此类具有较多数据的问题，Visual Basic 中引用了数组。在程序中使用数组的最大好处是用一个数组名代表逻辑上相关的一批数据，用下标表示该数组中的各个元素，与循环语句结合使用，使得程序书写简洁、结构清晰。

数组并不是一种数据类型，而是一组相同类型的变量集合，数组中的每个数据称为数组的元素，元素在数组中按线性顺序排列。用数组名代表逻辑上相关的一批数据，每个元素用下标变量来区分；下标变量代表元素在数组中的位置。高级语言中，可以定义不同维数的数组。所谓维数，是指一个数组中的元素需要用多少个下标变量来确定。一维数组相当于数学中的数列，二维数组相当于数学中的矩阵。

Visual Basic 中的数组，按不同的方式可分为以下几类：

(1) 按数组的大小(元素个数)是否可以改变可分为定长数组和动态(可变长)数组。

(2) 按元素的数据类型可分为数值型数组、字符串数组、日期型数组和变体数组等。

(3) 按数组的维数可分为一维数组、二维数组和多维数组。

(4) 按对象数组可分为菜单对象数组和控件数组。

5.1　数组的声明

5.1.1　一维数组的声明

Visual Basic 中数组没有隐式声明，所有使用的数组必须先声明，后引用。数组的声明包括声明数组名、数组的维数、每一维的元素个数及元素的数据类型。

一维数组的声明格式为：

Dim 数组名([<下界>to]<上界>)[As <数据类型>]

或

im 数组名[<数据类型符>]([<下界>to]<上界>)

例如：

```
Dim a(1 to 10) As Integer                    '声明了 a 数组有 10 个元素
```

与上面声明等价形式：Dim a%(1 to 10)

说明：

(1) 数组名的命名规则与变量的命名相同。

(2) 数组的元素个数，由它的“<下界>”和“<上界>”决定：上界 - 下界 + 1。

(3) 缺省“<下界>”为0,若希望下标从1开始,可在模块的通用部分使用Option Base语句将其设为1。其使用格式为:

```
Option Base 0|1          '后面的参数只能取0或1
```

例如:

```
Option Base 1            '将数组声明中缺省“<下界>”下标设为1
```

注意:该语句只能对本模块中声明时默认下界的数组起作用,对其他模块的数组不起作用。

(4) “<下界>”和“<上界>”不能使用变量,必须是常量,常量可以是直接常量、符号常量,一般是整型常量。其取值不能超过Long数据类型的范围(-2147483648～2147483647)。如果是实数,系统则自动按四舍五入取整。

(5) 如果省略As子句,则数组的类型为变体类型。

(6) 数组中各元素在内存中占一片连续的存储空间,一维数组在内存中存放的顺序是按下标大小的顺序。

5.1.2 一维数组的引用

对于数组必须先声明,后引用。对数组元素的操作与对简单变量的操作基本一样,在高级语言中一般只能逐个引用数组元素,而不能一次引用整个数组。Visual Basic 6.0中虽然可以将一个数组的内容赋值给另一个数组,但在实际操作中有许多注意事项,建议不要整体使用。

数组元素引用形式:数组名(下标)

其中:下标可以是整型变量、常量或表达式。

例如,设有下面的数组定义:

```
Dim A(15) As Integer,B(15) As Integer
```

则下面的语句都是正确的。

```
A(1)=A(2)+B(1)+5         '取数组元素运算
A(i)=B(i)                '下标使用变量
B(i+1)=A(i+2)            '下标使用表达式
```

注意:引用时下标不能超界(小于下界或大于上界),否则将出错。

5.1.3 二维数组的声明

二维数组的声明格式如下:

Dim 数组名([<下界> to] <上界>,[<下界> to]<上界>) [As <数据类型>]

其中的参数与一维数组完全相同。

例如:

```
Dim a(2,3) As Single
```

二维数组在内存的存放顺序是“先行后列”。例如,数组a的各元素在内存中的存放顺序是:a(0,0)→a(0,1)→a(0,2)→a(0,3)→a(1,0)→a(1,1)→a(1,2)→a(1,3)→a(2,0)→a(2,1)→a(2,2)→a(2,3)。

5.1.4 二维数组的引用

与一维数组一样,二维数组也是要事先声明后才能引用。

数组元素的引用形式为:

数组名(下标1,下标2)

例如,设有下面的数组定义:

Dim a(2,3) As Integer

则下面的语句都是正确的。

a(1,2) = 10

a(i+2,j) = a(2,3) * 2

在程序中常常通过二重循环来操作使用二维数组元素。

5.1.5 动态数组的建立

定长数组是指在声明时,就确定了数组的维数和大小的数组;动态数组是指在声明时未给出数组的大小,在程序执行时,再由 Redim 语句来确定维数及大小,并分配存储空间的数组。动态数组与定长数组的不同是:定长数组是在程序编译时分配存储空间的。

建立动态数组包括声明和大小说明两步:

(1) 使用 Dim、Private 或 Public 语句声明括号内为空的数组。其格式为:

Dim | Private|Public 数组名()[As <数据类型>]

(2) 在过程中用 ReDim 语句指明该数组的大小。其格式为:

ReDim [〈Preserve〉] 数组名([〈下界〉 to] 〈上界〉 [,[〈下界〉 to] 〈上界〉,…]) [As 数据类型]

例如:

```
Dim MyArray() As Integer          '声明动态数组
Redim MyArray(5)                  '在过程中定义数组为 5 个元素
Redim Preserve MyArray(1 to 15)   '在过程中定义数组为 15 个元素,保留数组中原数据
```

说明:

(1) ReDim 语句是一个可执行语句,只能出现在过程中,并且可以多次使用,改变数组的维数和大小。

(2) 定长数组声明中的"<下界>"和"<上界>"只能是常量,而动态数组 ReDim 语句中的"<下界>"和"<上界>"可以是常量,也可以是有确定值的变量。

(3) 可以使用 ReDim 语句反复地改变数组的元素个数及维数,但是不能在将一个数组定义为某种数据类型之后,再使用 ReDim 语句将该数组改为其他数据类型。

(4) 每次使用 ReDim 语句都会使原来数组中值丢失,可以在 ReDim 后加 Preserve 参数(可选)来保留数组中的数据。但如果使用 Preserve 参数,就只能重定义数组最末维的大小,不能改变数组的维数。例如,如果数组就是一维的,则可以重定义该维的大小,因为它是最末维,也是仅有的一维;对于二维或更多维时,则只有改变其最末维才能同时保留原数组中的内容。

例如,下面的二维动态数组增加第二维的大小,同时不清除所包含的任何数据。

```
Redim sum(10,10)
...
Redim Preserve sum(10,15)
```

5.2 数组的操作

5.2.1 数组操作语句和函数

1.数组清除语句

数组清除语句(Erase)可以作用于定长数组和动态数组,其格式为:

Erase 数组名[,数组名]…

功能:该语句用来清除定长数组的内容,或者释放动态数组占用的内存空间。

例如:

```
Dim array1(20) As Integer
Dim array2() As Single
Redim array2(9,10)
…
Erase array1,array2
```

说明:

(1) 对于定长数组,Erase 语句将数组重新初始化,即把所有数组元素置为 0(数值型)或空字符串(字符型)。

(2) 对于动态数组,Erase 语句将释放动态数组所占用的内存空间,也就是说,经 Erase 语句处理后,动态数组将不再存在。而定长数组经 Erase 语句处理后仍然存在,只是其内容被清零。

2. Array 函数

在 Visual Basic 中,当需要赋值的数组元素较多时,可以使用 Array 函数,其格式为:

数组变量名 = Array(数组元素值)

其中,"数组变量名"是预先定义的数组名,"数组元素值"是一个用逗号隔开的值表。

Array 函数用来为数组元素赋值,即把一组数据读入某个数组。

例如:

```
Dim D As Variant                    '定义数组名(变体类型)
D=Array(1,2,3,4)
```

执行上述语句后,将把 1,2,3,4,这 4 个数值赋给数组 D 的各个元素,即 1→D(0),2→D(1), 3→D(2), 4→D(3)。

说明:

(1) 数组变量名(如 D)后面不能有括号,也就没有维数和上界,下界默认为 0 或由 Option Base 语句决定。

(2) 只能给声明为 Variant 的变量或仅由括号括起的动态数组赋值。赋值后的数组大小由赋值的个数决定。

(3) Array 函数只能给一维数组赋值,不能给二维或多维数组赋值。通过 Array 函数给数组赋值后,就可以像使用一般数组一样来引用该数组。

3. UBound 函数和 LBound 函数

UBound()函数和 LBound()函数分别用来确定数组某一维的上界和下界值。

使用格式为:

UBound(<数组名>[, <n>])

LBound(<数组名>[, <n>])

其中:

"<数组名>"是必需的。指数组变量的名称,遵循标准变量命名约定。

"<n>"是可选的,一般是整型常量或变量。指定返回哪一维的上界或下界。1 表示第

一维,2 表示第二维,……。如果省略默认是 1。

例如,要打印动态数组 a 的各个元素值,可以通过下面的程序段来实现:

```
For i = Lbound(a) To Ubound(a)
  Print a(i)
Next i
```

4. Split 函数

使用 Split 函数可从一个字符串中,以某个指定符号为分隔符,分离若干个子字符串,建立一个下标从零开始的一维数组。

使用格式为:

Split(<字符串表达式> [,<分隔符>])

说明:

"<字符串表达式>"是必需的。是指包含子字符串和分隔符的字符串表达式。如果"<字符串表达式>"是一个长度为零的字符串(""),Split 函数则返回一个空数组,即没有元素和数据的数组。

"<分隔符>"是可选的。用于标识子字符串边界的字符。如果忽略,则使用空格字符(" ")作为分隔符。如果"<分隔符>"是一个长度为零的字符串,则返回的数组仅包含一个元素,即完整的"<字符串表达式>"字符串。

【例 5-1】 在一个文本框中,输入多个用","分隔的整数,按回车键后,将各数据按升序打印输出在窗体上。

解题思路:利用上述数组操作函数予以实现。在窗体上建立一个文本框 Text1,用以输入多个整数,程序代码写在该文本框的 KeyPress 事件中。程序运行界面如图 5-1 所示,程序代码如下:

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
  If KeyAscii = 13 Then
    Dim i%, j%, p%, n%, m%, t%
    Dim b() As Integer,a() As String
    '将文本框输入的文本,以","为分隔符分离字符子串存入数组 a 的各元素中
    '输入","时在英文状态下输入
    a = Split(Text1.Text,",")
    n = LBound(a):m = UBound(a)
    ReDim b(n To m)
    For i = n To m          '将数组 a 中的数字字符串转化成数字存入数组 b 中
      b(i) = Val(a(i))
    Next i
    For i = n To m - 1      '排序
      p = i
      For j = i + 1 To m
        If b(p) > b(j) Then p = j
      Next j
```

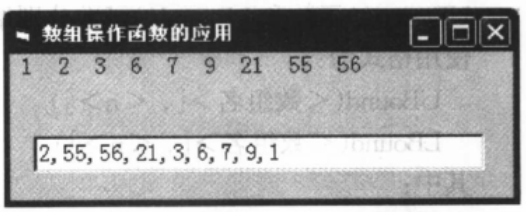


图 5-1 例 5-1 运行结果

```
        t=b(i):b(i)=b(p):b(p)=t
        Next i
        For i=n To m        '将排序后的数据输出到窗体上
          Print b(i);
        Next i
      End If
    End Sub
```

5.2.2 For Each…Next 语句

For Each…Next 语句针对一个数组或集合中的每个元素，重复执行一组语句，用一个变量来代表数组的每一个元素，该变量相当于数组或集合中的每一成员。对于集合来说，变量可以是变体型(Variant 型)变量或对象变量；对于数组而言，变量只能是变体型变量。其格式为：

```
For Each <变量 > In <数组名|对象集合>
  <循环体>
Next [ <变量> ]
```

说明：

只要数组或集合中有一个元素，就会进入循环体执行。针对数组或集合中的每一个元素执行一遍循环，直到处理完所有元素后，才会退出循环。

【例 5-2】 用 For Each…Next 语句，求 1! +2! +…+10! 的值。

解题思路：采用 Print 直接在窗体上输出结果，程序代码如下：

```
Private Sub Form_Load()
  Dim a(1 To 10) As Long, sum As Long, t As Long
  Dim n As Integer
  Show
  t=1
  For n=1 To 10
    t=t * n
    a(n)=t
  Next n
  sum=0
  For Each x In a
    sum=sum+x
  Next x
  Print "1! + 2! + 3! + …… 10! ="; sum
End Sub
```

程序运行结果为：1! + 2! + 3! + …… 10! =4037913

5.3 数组的应用举例

5.3.1 一维数组的应用举例

【例 5-3】 输入10名学生的成绩,求出最高分和最低分。

解题思路:创建应用程序的用户界面和设置对象属性。在窗体上建立 1 个标签(Label1)和 1 个命令按钮(Command1)。标签 Label1 用于显示有关信息,按钮 Command1 的 Caption 属性值为"查找"。

程序通过 Array 函数输入 10 个成绩,单击"查找"按钮(Command1)后,开始查找最高分和最低分,找到后显示在标签 Label1 上。程序运行界面如图 5-2 所示,程序代码如下:

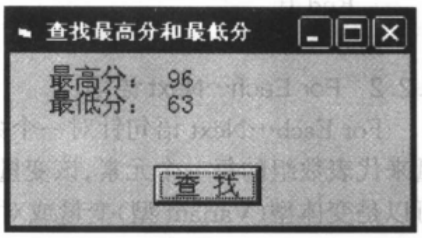


图 5-2 例 5-3 运行结果

```
Dim score As Variant                    '在窗体通用部分声明变量
Private Sub Form_Load()
  Label1.Caption="单击"查找"按钮开始查找最高分和最低分"
  score=Array(89,96,81,67,79,90,63,85,95,83)
End Sub
Private Sub command1_click()
  Dim max As Integer, min As Integer
  max=score(1)                          '设定初值
  min=score(1)
  For i=2 To 10
    If max<score(i) Then                '找最高分
      max=score(i)
    End If
    If min>score(i) Then                '找最低分
      min=score(i)
    End If
  Next i
  Label1.Caption="最高分:"+Str(max)+Chr(13) +"最低分:"+Str(min)
                                        'Chr(13)起换行作用
End Sub
```

5.3.2 二维数组的应用举例

【例 5-4】 编写程序:输出如下所示的杨辉三角形。

```
n=0  1
n=1  1  1
n=2  1  2  1
n=3  1  3  3  1
```

n=4　1　4　6　4　1
n=5　1　5　10　10　5　1
…

解题思路:杨辉三角形中的各行是二项式$(a+b)^n$展开式中的各项的系数,从上面的数据排列格式可以看出,可以用一个n×n方阵表示,方阵第1列上的元素都是1,对角线上的元素都是1,其余元素都是它上一行的前一列元素和同一列元素之和。

创建应用程序的用户界面和设置对象属性。在窗体上建立1个标签、1个文本框(Text1)和1个命令按钮(Command1)。文本框的MultiLine属性值置为True,用于显示生成的杨辉三角形;按钮Command1的Caption属性值为"根据输入的n值生成相应的杨辉三角形",单击该按钮会显示InputBox输入框,提示用户输入n的值,当输入完毕之后(例如n=5),将在文本框中显示与此n值相对应的杨辉三角形。运行界面如图5-3所示,程序代码如下:

```
Private Sub Command1_Click()
  n% = Val(InputBox("n=?"))
  Dim a&()
  ReDim a(n, n)
  For i=0 To n
    a(i, 0)=1:a(i,i)=1
  Next i
  For i=2 To n
    For j=1 To i-1
      a(i, j)=a(i-1,j-1)+a(i-1,j)
    Next j
  Next i
  For i=0 To n
      p$ =""
      For j=0 To i
        p=p & Format(a(i, j), "@@@@@@@@@@@@")
      Next j
      Text1.Text = Text1.Text & p & vbCrLf
  Next i
End Sub
```

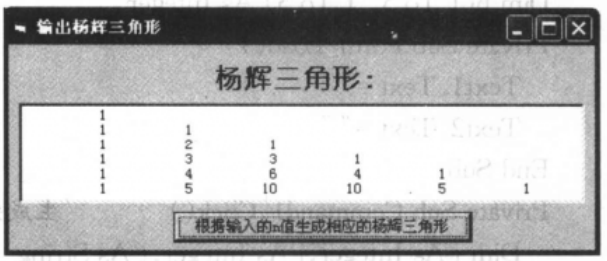


图5-3　例5-4运行结果

【例5-5】 编写程序:生成如下所示的矩阵A和B(B是A的转置矩阵),并分别在文本框中输出。

$$\mathbf{A}=\begin{bmatrix}1 & 3 & 5 & 7 & 9\\ 19 & 17 & 15 & 13 & 11\\ 21 & 23 & 25 & 27 & 29\end{bmatrix}\qquad \mathbf{B}=\begin{bmatrix}1 & 19 & 21\\ 3 & 17 & 23\\ 5 & 15 & 25\\ 7 & 13 & 27\\ 9 & 11 & 29\end{bmatrix}$$

解题思路:矩阵 A 生成的规则如下:

(1) 设 i 是行下标,j 是列下标。

(2) 当 i 为奇数时,a(i,j)=(i−1)×10+2×j−1

(3) 当 i 为偶数时,a(i,j)=i×10−2×j+1。

矩阵 A 和 B 互为转置矩阵,它们满足:a(i,j)=b(j,i)。其中,1≤i≤3,1≤j≤5。

创建应用程序的用户界面和设置对象属性。在窗体上建立 2 个标签、2 个文本框(Text1 和 Text2)、2 个命令按钮(Command1 和 Command2)。2 个文本框的 MultiLine 属性值为 True,"矩阵 A"文本框(Text1)用于显示矩阵 A,"矩阵 B"文本框(Text2)用于显示转置矩阵 B;按钮 Command1 的 Caption 属性值为"生成矩阵 A",单击该按钮可以生成矩阵 A,按钮 Command2 的 Caption 属性值为"生成转置矩阵 B",单击该按钮可以生成矩阵 B。程序的运行界面如图 5−4 所示,程序代码如下:

```
Dim a(1 To 3, 1 To 5) As Integer          '在窗体通用部分定义
Dim b(1 To 5, 1 To 3) As Integer
Private Sub Form_Load()
  Text1.Text=""
  Text2.Text=""
End Sub
Private Sub Command1_Click()              '生成矩阵 A
  Dim i As Integer,j As Integer,t As String
  t=""
  For i=1 To 3
    For j=1 To 5
      If i Mod 2=0 Then
        a(i, j)=i*10-2*j+1
      Else
        a(i, j)=(i-1)*10+2*j-1
      End If
      t=t & Format(a(i, j),"@@@")
    Next j
    t=t+vbCrLf
  Next i
  Text1.Text=t
End Sub
Private Sub Command2_Click()              '生成转置矩阵 B
  Dim i As Integer,j As Integer,t As String
  t=""
  For i=1 To 5
    For j=1 To 3
      b(i,j)=a(j,i)
```

图 5−4 例 5−5 运行结果

```
            t=t & Format(b(i, j), "@@@")
        Next j
        t=t+vbCrLf
    Next i
    Text2.Text=t
End Sub
```

5.3.3　动态数组的应用举例

【例5-6】　动态数组的应用示例。

解题思路：采用Print直接在窗体上输出结果，程序运行界面如图5-5所示，程序代码如下：

```
Private Sub Form_Load()
    Dim a() As Integer
    Show
    ReDim a(800)
    k=0
    For x=100 To 600 Step 7
        If x Mod 8=0 Then
            k=k+1
            a(k)=x
        End If
    Next x
    ReDim Preserve a(k)
    For j=1 To k
        Print a(j)
    Next j
End Sub
```

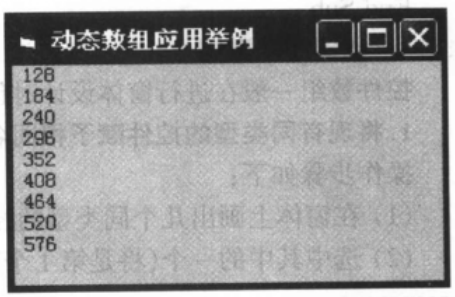


图5-5　例5-6运行结果

5.4　控件数组

在设计应用程序界面时，常把具有相同类型和功能相似的多个控件设计为数组，称为控件数组，这些控件具有以下特点：

(1) 具有相同的控件名称(即Name属性相同，都为控件数组名)，并以下标索引号(Index，相当于一般数组的下标)来识别各个控件。每一个控件称为该控件数组的一个元素，表示为：

控件数组名(索引号)

控件数组至少应有一个元素，最多可达32 767个元素。第一个控件的索引号默认为0，也可以是一个非0的整数。Visual Basic允许控件数组中控件的索引号不连续。

(2) 具有相同的一般属性。

(3) 共用相同的事件过程。控件数组的事件过程会返回一个索引号(Index)，以确定当

前发生该事件的是哪个控件。

例如,在窗体上建立一个命令按钮数组 Command1,运行时不论单击哪一个按钮,都会调用以下事件过程。在程序运行中,可以利用返回控件的索引号(Index)来识别事件是由哪个控件引发的,常用 Select Case Index 语句表示引发不同控件所产生的不同动作。具体程序代码如下:

```
Private Sub Command1_Click(Index As Integer)
  …
  Select Case Index          '根据 Index 值的不同执行不同的程序代码
    Case 0                   '单击 Command1(0)的程序代码
      …
    Case 1                   '单击 Command1(1)的程序代码
      …
  End Select
  …
End Sub
```

5.4.1 创建控件数组

控件数组一般在进行窗体设计时创建,常用如下几种方法。

1. 将现有同类型的控件赋予相同名称(Name)创建

操作步骤如下:

(1) 在窗体上画出几个同类型的控件;

(2) 选中其中的一个(将是第 1 个数组元素)控件,将它的 Name 属性改为新名字,例如 samename;

(3) 选中第 2 个控件,也将它的 Name 属性改为相同的新名字,即 samename;

(4) 在图 5-6 弹出的对话框内,选择"是"按钮,这时就创建了控件数组 samename,它包含两个元素 samename(0)和 samename(1);

(5) 再选中另一个控件,将它的 Name 属性也改为相同的名字,即 samename;

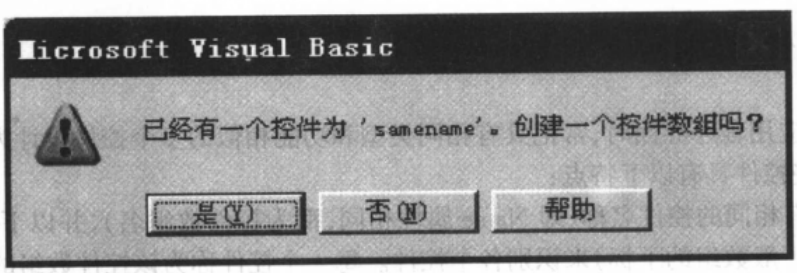


图 5-6 创建控件数组

(6) 重复步骤(5),这样就可以在该控件数组中添加更多的控件。

2. 复制现存控件创建

操作步骤如下:

(1) 在窗体上画出控件数组的第 1 个控件;

(2) 选中所画的控件,按 Ctrl+C 组合键进行复制,选定目标位置,按 Ctrl+V 组合键进

行粘贴；

(3) 如果出现类似于图 5－6 弹出的对话框，选择“是“按钮；

(4) 重复步骤(2)，就可以在该控件数组中添加更多的控件，这样复制的各个控件的属性是一致的(只有 Index 属性不同)，通常还需要分别设置某些属性。

3. 设置控件的 Index 属性来创建

操作步骤如下：

(1) 在窗体上画出控件数组的第 1 个控件，将它的 Index 属性设置为 0(原来为空值)；

(2) 选中所画的控件，按 Ctrl＋C 组合键进行复制，选定目标位置，按 Ctrl＋V 组合键进行粘贴，此时不会出现对话框，就已经在控件数组中添加了 1 个控件；

(3) 多次按 Ctrl＋V 组合键进行粘贴，就可以在该控件数组中添加更多的控件。

4. 使用 Load 语句添加控件数组的新控件成员

可以在程序运行中使用 Load 语句加载该控件数组的新控件成员。Load 语句的格式为：

Load〈数组名(下标值)〉

功能：在以“〈数组名〉”为名称的控件数组已经存在的情况下，可以使用 Load 语句加载该控件数组的新控件成员，所给的下标值就是该新成员的下标。

【例 5－7】 使用控件数组创建一个标题为“项目 1“、“项目 2”、“项目 3”、……、“项目 6”的复选框组，并将该复选框数组放置在标题为“项目选择器”的框架容器内。

解题思路：首先在窗体的适当位置画一个框架，设置标题为“项目选择器”，再在该框架内画一个复选框，并将它的 Index 属性设置为 0，然后在窗体的加载事件过程中编写如下代码，程序运行结果如图 5－7 所示。

```
Private Sub Form_Load()
  With Frame1
    .Left = 120
    .Top = 120
    .Width = 6500
    .Height = 1000
  End With
  With Check1(0)
    Check1(0).Caption = "项目 1"
    Check1(0).Left = 400
    Check1(0).Top = 300
    Check1(0).Width = 800
    Check1(0).Height = 400
  End With
  For n = 1 To 5
    Load Check1(n)
    With Check1(n)
      .Left = Check1(0).Left + n * 1000
      .Visible = True
```

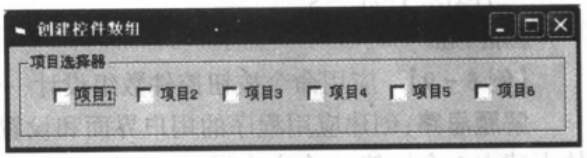


图 5－7　例 5－7 运行结果

```
        .Caption = "项目" & n + 1
      End With
    Next n
  End Sub
```

说明：在程序中 3 次使用 With…End With 语句，它可以对同一对象执行多个不同的操作。例如，这里对同一个对象 Check1(0)设置 5 个属性，避免重复书写“Check1(0)”，简化了程序。

5.4.2 控件数组的应用举例

【例 5-8】 控件数组的应用示例。

解题思路：按图 5-8 设计窗体，窗体左端 4 个文本框构成一控件数组，从上到下 Index 属性值为 0,1,2,3，即 Name 属性值从上到下依次为 Text1(0)、Text1(1)、Text1(2)和 Text1(3)，该控件数组在程序运行时用于输入数据；窗体的右上端放置一“输出总数”文本框(Text2)，该文本框在程序运行时用于显示左端控件数组输入的数据之和；窗体的右下端放置一命令按钮(Command1)，其 Caption 属性值为“计算总数”，在程序运行时，单击该按钮可实现在“输出总数”文本框中显示控件数组中输入的数据之和。程序代码如下：

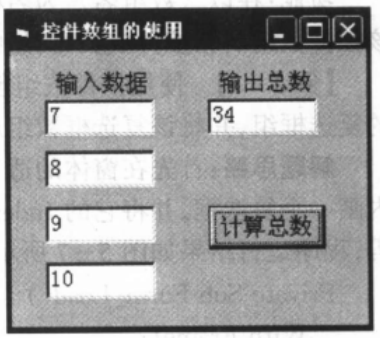


图 5-8 例 5-8 运行界面

```
Private Sub Command1_Click()
  s = 0
  For i = 0 To 3
    s = s + Val(Text1(i).Text)
  Next i
  Text2.Text = s
End Sub
```

【例 5-9】 应用命令按钮控件数组设计一个无键盘情况下的“会员登记窗”。

解题思路：创建应用程序的用户界面和设置对象属性，程序界面如图 5-9 所示。在窗体上建立 1 个标签，1 个文本框(Text1)，用于显示用户输入的会员号；1 个框架(Frame1)，其 Caption 属性值为“无键盘操作框：”；1 个命令按钮控件数组，所有按钮均放在框架内。程序运行时，当单击 0～9 数字按钮时，将在文本框中逐个显示所按下的数字；单击“确认”按钮用于确认所输入的会员号，并且在该窗口将显示用户先前输入的会员号信息；单击“清除”按钮用于清除文本框。程序代码如下：

```
Private Sub Command1_Click(Index As Integer)
  Select Case Index
    Case 0 To 9
      Text1.Text = Text1.Text & Command1(Index).Caption
    Case 10
      Frame1.Visible = False
      Text1.Visible = False
      Label1.Caption = "你的会员号是：" & VbCrLf & Text1.Text
    Case 11
```

```
        Text1.Text =""
    Case 12
        Unload Me
  End Select
End Sub
```

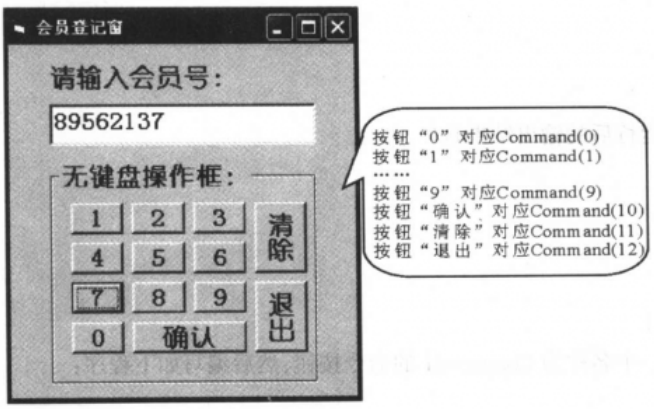


图5-9　例5-9运行界面

习题五

1.简答题

(1) 定长数组与动态数组的主要区别是什么？使用动态数组有什么优点？

(2) 一维和二维数组在内存中的存储方式是什么样的？

(3) 如何通过循环语句针对数组进行操作？For…Next型循环和For Each…Next型循环有什么不同？

(4) 如何创建控件数组？使用控件数组有哪些优势？

(5) 简述数组的相关操作语句和函数的基本含义及注意事项。

(6) 简述使用动态数组的注意事项。

2.填空题

(1) 控件数组的名字由(　　)属性指定，数组元素的下标由(　　)属性指定。

(2) 由Array语句进行初始化的数组必须定义为(　　)类型。

(3) 如果定义一个数组“Dim D(5) As Integer”，其元素最多有(　　)个，如果之前在窗体放置语句“Option Base 1”，则元素的个数有(　　)个。

(4) 以下程序段运行后的输出结果是(　　)。

```
Dim a(5) As Integer
For i=1 To 5
  a(i)=i*i
Next i
Print a(i - 1)
```

(5) 设有声明语句：

```
Option Base 0
Dim B(-1 To 10,2 To 9,20) As Integer
```

则数组 B 中全部元素的个数为(　　)。

(6) 以下程序段运行后的输出结果是(　　)。

```
Dim n()
n=Array(1,2,3)
k=2
For i=0 To 2
  n(i)=n(i)+1
Next i
Print n(k)
```

(7) 以下程序段运行后的输出结果是(　　)。

```
Dim a(100)
For i=1 To 100
  a(i)=2 * i
Next i
Print a(a(24)-1)
```

(8) 在窗体上画一个名称为 Command1 的命令按钮,然后编写如下程序:

```
Option Base 1
Private Sub Command1_Click()
  Dim c As Integer, d As Integer
  d=0
  c=6
  x=Array(2,4,6,8,10,12)
  For i=1 To 6
    If x(i)>c Then
      d=d+x(i)
      c=x(i)
    Else
      d=d-c
    End If
  Next i
  Print d
End Sub
```

程序运行后,如果单击命令按钮,则在窗体上输出的内容为(　　)。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

3. 选择题

(1) 下面的数组声明语句中,正确的是(　　)。

A. Dim MA(1,5) As String　　B. Dim MA(1 To 5,2 To 5) As String

C. Dim MA(2 To 5) As String　　D. Dim MA(1:5,2:5) As String

(2) 以下说法不正确的是(　　)。

A. 使用 ReDim 语句可以改变数组的维数

B. 使用 ReDim 语句可以改变数组的类型

C. 使用 ReDim 语句可以改变数组的每一维的大小

D. 使用 ReDim 语句可以改变对数组中的所有元素进行初始化

(3) 使用语句 Dim A As Integer 声明数组 A 之后,以下说法正确的是(　　)。

A. A 数组中所有元素值为 0

B. A 数组中所有元素值不确定

C. A 数组中所有元素值为空

D. 执行 Erase A 后,A 数组中的所有元素值不为 0

(4) 假如用复制、粘贴的方法建立一个命令按钮数组 Command1,以下对该数组的说法错误的是(　　)。

A. 命令按钮的所有 Caption 属性都是 Command1

B. 在代码中访问命令按钮只需使用名称 Command1

C. 命令按钮的大小都相同

D. 命令按钮共享相同的事件过程

(5) 阅读程序:

```
Option Base 1
Dim arr() As Integer
Private Sub Form_Click()
  Dim i As Integer,j As Integer
  ReDim arr(3,2)
  For i=1 To 3
    For j=1 To 2
      arr(i,j)=i*2+j
    Next j
  Next i
  ReDim Preserve arr(3,4)
  For j=3 To 4
    arr(3, j)=j+9
  Next j
  Print arr(3, 2)+arr(3,4)
End Sub
```

程序运行后,单击窗体,输出结果为(　　)。

A. 21　　B. 13　　C. 8　　D. 25

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(6) 下列程序段的执行结果为(　　)。

```
Dim m(5,6), s(5)
For i=1 To 5
  s(i)=0
  For j=1 To 5
    m(i, j)=i+j
    s(i)=m(i,j)+2*i+j
  Next j
Next i
For Each x In s
  Print x;
Next x
```

A. 13　16　19　22　25　　B. 20　25　30　35　40

C. 7 10 13 16 19　　D. 13 17 20 23 26

(7) 下列程序段的执行结果为(　　)。

```
Dim b(-2 To 5)
For i=LBound(b,1) To UBound(b,1)
  b(i)=i*i
Next i
Print b(LBound(b,1)); b(UBound(b,1))
```

A. 4 25　　B. -2 5　　C. 0 0　　D. 1 25

(8) 运行下列程序会出现错误信息,产生错误的原因是(　　)。

```
x=5
Dim a(x)
For i=1 To 6
  a(i)=i+1
Next i
```

A. 数组元素 a(i)的下标越界　　B. 变量 x 没有定义

C. 循环变量的范围越界　　D. Dim 语句中不能用变量 x 来定义数组的下标

(9) 在窗体上画一个名称为 Label1 的标签,然后编写如下事件过程:

```
Private Sub Form_Click()
  Dim arr(10,10) As Integer
  Dim i As Integer, j As Integer
  For i=2 To 4
    For j=2 To 4
      arr(i, j)=i * j
    Next j
  Next i
  Label1.Caption=Str(arr(2,2)+arr(3,3))
End Sub
```

程序运行后,单击窗体,在标签中显示的内容是(　　)。

A. 12　　B. 13　　C. 14　　D. 15

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(10) 下列程序段的执行结果为(　　)。

```
Dim a(8) As Integer
y=18
i=0
Do
  a(i)=y Mod 2
  i=i+1
  y=y\2
Loop While (y>=1)
For j=i-1 To 0 Step-1
  Print a(j);
Next j
```

A. 1 0 0 0　　B. 1 0 0 1 0　　C. 0 0 1 1 0　　D. 1 0 1 0 0

4. 编程题

(1) 从键盘输入10个数据保存在数组A中，找出其中的最大数与最小数并将其对调，将对调前后的数组元素分别在两个文本框中输出。

(2) 设计程序，随机产生64个100～999范围内的整数，存放在8×8的数组中，然后找出该数组中最大值的元素(若有多个最大值元素，只需找出其中一个)，并输出其值和所在的行号和列号(行号和列号都从1算起)。

(3) 在窗体WinForm1中建立1个单选按钮组Opt1和1个命令按钮Cmd1，单选按钮组包含3个单选按钮，其标题分别为“单选1”、“单选2”和“单选3”，下标分别为0、1和2。初始状态下，“单选2”单选按钮被选中，以后每次单击命令按钮时，依次选中1个单选按钮。

第 6 章　过程

在前面的章节中，基本上是使用系统提供的事件过程和内部函数进行程序设计。事实上，Visual Basic 允许用户定义自己的过程和函数。用户使用自定义过程和函数，不仅能够提高代码利用率，而且使得程序结构更为清晰、简洁，便于调试和维护。

6.1　过程的定义与分类

6.1.1　过程的定义

在设计程序时，有些程序代码常常需要重复执行，或者许多个程序都要进行同类操作。这些重复执行的程序是相同的，只是每次都以不同的参数进行重复。例如，判断整数 N1 和 N2 是否是素数的两段程序几乎完全相同。若将判断任意整数 N 是否是素数写成一个返回逻辑值的函数，把要判断的数每次作为参数进行传递，那么程序结构就变得简单清晰。

通常将程序分割成较小的、完成一定任务的、相对独立的程序段（通常称为逻辑部件），以简化程序设计，此时称这些程序段为过程。过程可以变成增强和扩展 Visual Basic 的构件，过程的使用可以较好地压缩重复程序和共享文件。其优点有两点：

(1) 过程可使程序划分成离散的逻辑单元，每个单元都比无过程的整个程序容易调试；

(2) 一个程序中的过程，往往不必修改或稍做改动，便可以成为另一个程序的构件。

一个过程程序仍然由顺序、选择、循环三种基本结构所成，但它却有自己的特点，主要体现在主程序与子程序之间的数据输入、输出，即主程序与子程序之间的数据传递。

6.1.2　过程的分类

在 Visual Basic 中，过程基本分成子程序过程、函数过程、属性过程和事件过程四类。

1. 子程序过程

子程序过程又称为自定义过程（也称为通用过程）。子程序过程用于完成特定的功能，必须由应用程序来调用，一般采用 Call 语句进行调用。建立子程序过程的原因是，几个不同的事件过程也许要执行同样的操作，或者一段程序具有相对独立的功能，用子程序过程完成对这一操作和功能的描述，并由事件过程调用它，不必重复编写代码，便于共享。调用发生在两个过程之间，通常将调用其他过程的过程称为主调过程，把被调用的过程称为被调过程。

2. 函数过程

函数过程又称为 Function 过程，是被调用后要返回值的过程。函数和子程序过程一样，也是具有独立功能的过程，因主调过程的调用而被执行，也能传递参数，执行一系列的语句，执行完毕后返回主调过程。函数无论从定义的格式上还是执行的方式上，都与子程序过程非常相似。

但二者本质的不同是被调用后能否向主调过程返回一个值。函数过程将在过程体中以“〈函数名〉=〈返回值〉”的形式向调用点返回一个结果值,并参与调用点所在环境的操作,函数返回值的使用特点与表达式的使用特点相似。

3.属性过程

属性过程用于返回和设置对象属性的值,还可以设置对属性的引用,并创建和引用用户自定义的属性,通过创建对象及属性来扩充 Visual Basic 的功能,属于较高层次的编程技术。

4.事件过程

事件过程是指附加在窗体和控件上的过程。在 Visual Basic 中的对象对一个事件的发生认定时,自动用对应于该对象和事件的名字调用该事件过程。

Visual Basic 中又把子程序过程和事件过程统称为子过程或 Sub 过程。子过程是指被调用后不返回值的过程,调用子过程相当于只把其中的语句段执行一遍就退出,不返回特定的值。

子程序过程和事件过程在大多数方面都具有子过程的共性,但二者有以下不同:

(1) 调用方式不同:子程序过程是在其他过程中通过调用语句 Call 实现调用的,有比较明显的调用语句;而事件过程是对一个对象触发某个事件时发生调用的,是在程序执行中根据用户的操作来触发事件过程的调用。

(2) 过程名的取名方法不同:子程序过程名完全由用户自己定义,而事件过程名是根据触发它的对象和事件的名称决定的,可以自动形成。

6.2 Sub 过程

在 Visual Basic 中有两类 Sub 过程:事件过程和自定义过程(又称子程序过程或通用过程)。

6.2.1 事件过程

在 Visual Basic 中的对象对一个事件的发生做出认定时,便自动用相应的事件名调用该事件的过程。事件名在对象和代码之间建立一种联系,事件过程是附加在窗体和控件上的程序。

一个控件的事件过程将控件的实际名字(在 Name 属性中规定的)、下划线(_)和事件名组合起来。例如,如果单击一个名为 Command1 的命令按钮,则会触发该事件过程,过程名为 Command1_Click。

一个窗体事件过程将单词“Form”、下划线和事件名组合起来。例如,如果单击窗体,则会触发单击窗体事件过程,过程名为 Form_Click。

1.控件事件过程的语法格式

```
Private Sub 控件名_事件名(参数列表)
  <语句组>
End Sub
```

2.窗体事件过程的语法格式

```
Private Sub Form_事件名(参数列表)
  <语句组>
```

End Sub

其中：<语句组>就是程序设计者编写的让该事件发生后完成相应操作的程序代码。

注意：使用 Visual Basic 提供的代码过程会自动地将过程名生成出来。在“代码编辑器”窗口中从“对象框”中选择一个对象，从“事件框”中选择一个事件，就可在窗口中生成一个模板。例如，当对象选为窗体 Form，事件选为 Click，则在“代码编辑器”窗口中生成如下模板：

Private Sub Form_Click()

End Sub

6.2.2 自定义过程

当几个不同的过程要执行同样的程序段，为了不必重复编写代码，可以采用自定义过程来实现。自定义过程只有在被调用时才起作用，它一般由事件过程来调用。自定义过程可以保存在窗体模块(.Frm)和标准模块(.Bas)中。

1. 自定义过程的定义

自定义过程的定义形式如下：

[Public|Private][Static] Sub 过程名([形式参数列表])
  <局部变量或常数定义>
  <语句组>
  [Exit Sub]
  <语句组>
End Sub

说明：

(1) 自定义过程以 Sub 开头，以 End Sub 结束，在 Sub 和 End Sub 之间是描述过程操作的语句块，称为“过程体”或“子程序体”。

(2) 过程名：命名规则与变量名规则相同。过程名不返回值，而是通过形式参数与实际参数的传递得到结果，调用时可返回多个值，实际参数通常简称“实参”。

(3) 形式参数列表：形式参数通常简称“形参”，形式参数列表仅表示形参的类型、个数、位置，定义时是无值的，只有在过程被调用时，形参与实参结合后才获得相应的值。过程可以无形式参数，但括号不能省。

形式参数列表定义形式如下：

[ByVal|ByRef]形参名[()][As 数据类型][,…],…

ByVal 表示当该过程被调用时，参数是按值传递的；ByRef 表示当该过程被调用时，参数是按地址传递的，默认为 ByRef。形参可以是简单变量，也可以是数组。

(4) 可选项“[Public|Private][Static]”的含义和作用如下：

① 如果选用 Private(局部的)，只有该过程所在模块(如窗体模块)中的过程才能调用该过程；如果选用 Public(公用的)，表示在应用程序中的任何地方都可以调用该过程。系统默认为 Public。

② 如果选用 Static，表示过程中的局部变量是静态变量，在过程被调用后，其值仍然保留；如果不用 Static 属性，则局部变量是动态的，每次调用过程时，局部变量的初始值为零值或空字符。

(5)"[Exit Sub]"表示退出过程。

在窗体的"代码编辑器"窗口中或标准模块的代码窗口中直接按定义形式输入建立过程的操作,也可以通过选择"工具"菜单中的"添加过程"命令实现。

选择"工具"菜单中的"添加过程"命令,出现"添加过程"对话框,如图 6-1 所示。选择过程类型(子程序、函数、属性、事件)及作用范围(公有的 Public、私有的 Private),单击确定后得到一个过程或函数定义的结构框架(模板),如下面代码:

```
Public Sub exchange()

End Sub
```

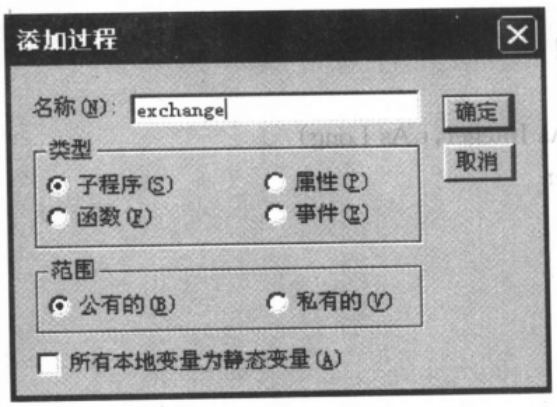


图 6-1　"添加过程"对话框

如果需要形参,则必须在括号中填写形参的定义。过程中的代码当然要由用户根据问题的要求写入。

2. 自定义过程的调用

Visual Basic 中自定义过程的调用有两种方法,具体形式为:

过程名 [实际参数列表]

或

Call　过程名[(实际参数列表)]

例如:

```
Call SubCal(10)
SubCal 10
```

说明:

(1) 用过程名调用时必须省略参数两边的括号。使用 Call 语句调用时,参数必须在括号内,当没有实参时,此时括号可以省略。

(2) 实参必须与形参保持个数相同,顺序与类型一一对应。

(3) 实参可以是变量、常数、表达式和数组。如果希望传递地址,实参应是变量或数组。

(4) 调用时把实参值传递给对应的形参。其中值传递(形参前有 ByVal 说明)时实参的值不随形参的值变化而改变。而地址传递(形参前有 ByRef 说明)时实参的值随形参值的改变而改变。

【例 6-1】 计算 5! + 10!。

解题思路:因为计算 5! 和 10! 都要用到阶乘 n! (n! = 1×2×3×…×n),所以把计算 n! 编成自定义过程。采用 Print 直接在窗体上输出结果,程序代码如下:

```
Private Sub Form_load()
  Dim y As Long, s As Long
  Show
  Call jc(5,y)
  s=y
  jc 10,y
  s=s+y
  Print "5! +10! =";s
End Sub
Private Sub jc(n As Integer,t As Long)
  Dim i As Integer
  t=1
  For i=1 To n
    t=t*i
  Next i
End Sub
```

程序的运行结果为:5! +10! =3628920。

根据过程的调用关系,如果过程 A 调用过程 B,将过程 A 称为主调过程,过程 B 称为被调过程。主调过程与被调过程是相对的,一般来说,一个过程既可作为主调过程又可作为被调过程。图 6-2 表示例 6-1 中过程调用的执行过程。

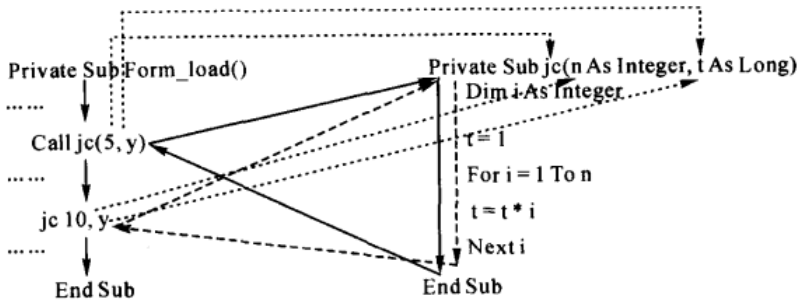


图 6-2 例 6-1 过程中调用的执行过程

在图 6-2 中,主调过程 Form_load 第一次调用自定义过程 jc 时,从第一个调用语句"Call jc(5, y)"处转到执行过程 jc,在过程 jc 中从第一条 Sub 语句开始执行过程 jc 中的程序代码,当执行到过程 jc 的 End Sub 语句时,便返回到主调过程中调用本次自定义过程的下一条语句,并由此继续执行主调过程。从"Call jc(5, y)"语句开始的调用与返回如图 6-2 中的实箭头线所示。当程序流程执行主调过程 Form_load 中自定义过程调用语句"jc 10, y"时,程序流程再次转去执行自定义过程 jc,…,从过程 jc 的 End Sub 语句返回到主调过程中调用本次自定义过程的下一条语句,再继续执行主调过程中的其他语句,如图 6-2 中的

长虚箭头线表示，图 6－2 中细虚箭头线表示参数的传递。

6.3 函数过程

Visual Basic 函数分为内部函数和外部函数。内部函数是系统预先编好的、能完成特定功能的一段程序，如 Sqr、Sin、Len 等。外部函数是用户根据需要用 Function 关键字定义的函数过程，与 Sub 过程不同的是函数过程将返回一个值。函数过程又称为 Function 过程。

6.3.1 函数过程的定义

和 Sub 过程一样，函数过程也是用来执行一个特定任务的独立程序代码。与 Sub 过程不同的是，函数过程返回一个值到调用的表达式中。

函数过程的定义形式如下：

```
[Public|Private][Static]Function 函数名([<形式参数列表>])[As<数据类型>]
  <局部变量或常数定义>
  <语句块>
  [函数名=返回值]
  [Exit Function]
  <语句块>
  [函数名=返回值]
End Function
```

说明：

(1) 函数名命名规则与变量命名规则相同。并且不能与系统的内部函数或其他通用子过程同名，也不能与已定义的全局变量和本模块中模块级变量同名。

(2) 在函数体内，函数名可以当变量使用，函数的返回值就是通过对函数名的赋值语句来实现的，即函数值通过函数名返回，因此在函数过程中至少要对函数名赋值一次。

(3) “As<数据类型>”是指函数返回值的类型，若省略，则函数返回变体类型值(Variant)。

(4) “[Exit Function]”表示退出函数过程，常常是与选择结构(If 或 Select Case 语句)联用，即当满足一定条件时，退出函数过程。

(5) 函数过程形参的定义与自定义过程完全相同。

(6) 可选项“[Public|Private][Static]”的含义和作用与自定义过程完全相同。

6.3.2 函数过程的调用

1.直接调用

直接调用方法和调用 Visual Basic 内部函数过程(如 Len)的方法一样，只需写出函数名和相应的参数即可。具体调用格式为：

函数名(实际参数列表)

例如：

```
S=Max(a,b)
```

2.用 Call 语句调用

与调用自定义过程一样调用函数过程。

用 Call 语句调用方法

例如：

```
Call Max(a,b)
```

当用这种方法调用函数过程时，将会丢弃返回值。

说明：

在调用时，实参和形参的数据类型、顺序、个数必须匹配。函数调用的功能是求得函数的返回值。

【例 6-2】 判断一个给定的整数是否是素数。

解题思路：素数是指除被 1 和自身整除外，不能被其他整数整除的自然数。判断整数 n 是不是素数的基本方法是：将 n 分别除以 2,3,…,n-1，若都不能整除，则 n 为素数。事实上不必除那么多次，因为 n=Sqr(n) * Sqr(n)，所以，当 n 能被大于等于 Int(Sqr(n))的整数整除时，一定存在一个小于等于 Int(Sqr(n))的整数，使 n 能被它整除。因此，只要判断 n 能否被 2,3,…,Int(Sqr(n))整除即可。判断 n 被 i 整除可用表达式 n mod i=0 或 n/i=n \ i 或 n/i=Int(n/i)表示。

创建应用程序的用户界面和设置对象属性，程序界面如图 6-3 所示。在窗体上建立 2 个标签，2 个文本框(Text1 和 Text2)，1 个命令按钮(Command1)，其 Caption 属性值为“判断”。“输入”文本框(Text1)在程序运行时用于输入一个整数，“显示”文本框(Text2)在程序运行时用于显示该整数是否为素数的信息，单击命令按钮，将会对输入的整数是否是素数进行判断。程序中判断一个整数是否是素数是用函数 Prime 实现的，具体代码如下：

```
Private Function Prime(n As Integer) As Boolean
  Dim k%,Yes As Boolean
  Yes=True
  For k=2 To Int(Sqr(n))
    If n Mod k=0 Then Yes=False:Exit For
  Next k
  Prime=Yes              '给函数名赋值,作为函数的返回值
End Function
Private Sub Command1_Click()
  Dim a As Integer
    a=Val(Text1.Text)
    If Prime(a) Then
      Text2.Text=a & "是素数"
    Else
      Text2.Text = a & "不是素数"
    End If
End Sub
```

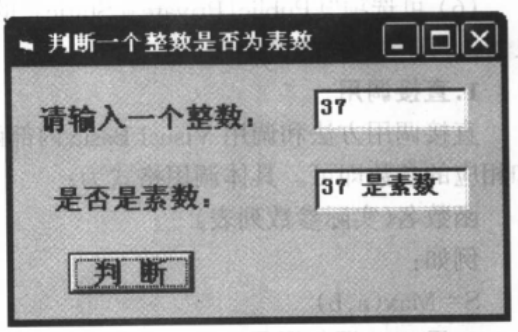


图 6-3 例 6-2 运行界面

6.4 过程参数的传递

Visual Basic 中不同模块(过程)之间数据的传递有两种方式:

(1) 通过过程调用实参与形参的结合实现;

(2) 使用全局变量来实现各过程中共享数据。

6.4.1 形式参数与实际参数

1.形式参数

形式参数通常简称为"形参",是指在定义过程时,出现在 Sub 或 Function 语句中的函数名后面圆括号内的变量名或数组名,用来接收或传送给过程的数据,形参表中的各个变量之间用逗号分隔。

2.实际参数

实际参数通常简称为"实参",是指在调用 Sub 或 Function 过程时,写入过程名或函数名后括号内的参数,其作用是将它们的数据(数值或地址)传送给 Sub 或 Function 过程与其对应的形参变量。

实参可由常量、表达式、有效的变量名、数组名(后加左、右括号,如 A())组成,实参表中各参数用逗号分隔。

6.4.2 参数传递的方式

参数传递是指将主调过程的实参(调用时已有确定值和内存地址的参数)传递给被调过程的形参。参数的传递有两种方式:按值传递、按地址传递。形参前加"ByVal"关键字的是按值传递,缺省或加"ByRef"关键字的为按地址传递。

(1) 按值传递:将实参的数值传递给过程中对应的形参变量。按值传递参数时,Visual Basic 给传递的形参分配一个临时的内存单元,将实参的值传递到这个临时单元中。

按值传递是单向的,如果在被调用的过程中改变形参变量的值,则只是临时单元的值变动,不会影响实参变量本身。当被调过程结束,返回主调过程时,Visual Basic 将释放形参变量的临时内存单元,实参变量的值不变。

(2) 按地址传递:形参前没有 ByVal 关键字或用 ByRef 关键字说明的,是一种将实参数据的地址传递给过程中对应形参变量的方式,形参变量和实参变量具有相同的地址,即形参和实参共用一组存储单元。如果在被调过程中改变形参变量的值,则返回后相应实参变量的值也被改变。因此,按地址传递是双向传递。利用这一特点,可以通过传地址的形参返回数据。

下面通过具体的例子说明两种传递方式。

【例 6-3】 使用"按值"传递方式编写交换两个变量 a 和 b 的自定义过程 Swap1。在 Command1_Click()事件过程中调用 Swap1 过程,并输出调用后的结果。

解题思路:程序运行界面如图 6-4 所示。在窗体上建立 6 个标签,4 个文本框(上面 2 个从左到右分别为 Text1, Text2,用来输入 a 和 b 交换前的值;下面两个从左到右分别为 Text3, Text4,用来显示 a 和 b 交换后的值),1 个命令按钮 Command1,其 Caption 属性值为"交换",单击此按钮执行过程 Swap1 的代码。程序代码如下:

```
Public Sub Swap1(ByVal x As Integer, ByVal y As Integer)    '形参定义为传值方式
```

```
    Dim t As Integer
    t=x:x=y:y=t
End Sub
Private Sub Command1_Click()
    Dim a As Integer, b As Integer
    a=Val(Text1.Text)
    b=Val(Text2.Text)
    Swap1 a, b                    '调用子过程 Swap1
    Text3.Text=a
    Text4.Text=b
End Sub
```

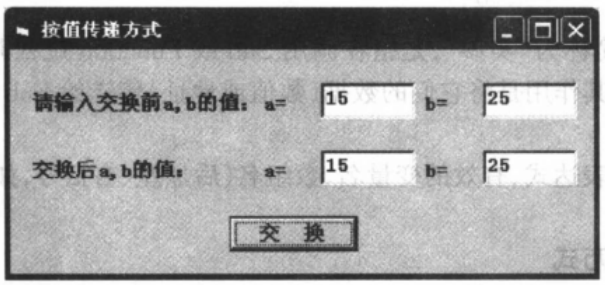


图 6-4 例 6-3 运行界面

分析:从输出结果可见,调用子过程 Swap1 没能交换变量的值,其原因是:过程 Swap1 采用值传递方式,过程被调用时,系统给形参分配临时内存单元 x 和 y,将实参 a 和 b 的值分别传递给 x 和 y[如图 6-5(b)];在过程 Swap1 中,变量 a、b 不可使用,x 和 y 通过临时变量 t 实现交换[如图 6-5(c)];调用结束返回主调过程后,形参 x 和 y 的临时内存单元将释放,实参单元 a 和 b 仍保留原值[如图 6-5(d)]。值传递的整体执行过程如图 6-5 所示。

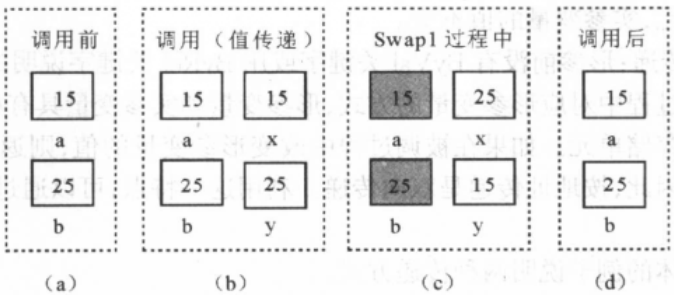


图 6-5 值传递执行过程

【例 6-4】 使用"按地址"传递方式编写交换两个变量 a 和 b 的自定义过程 Swap2。在 Command1_Click()事件过程中调用 Swap2 过程,并输出调用后的结果。

解题思路:程序运行界面如图 6-6 所示。在窗体上建立 6 个标签,4 个文本框(上面 2 个从左到右分别为 Text1, Text2,用来输入 a 和 b 交换前的值;下面 2 个从左到右分别为 Text3, Text4,用来显示 a 和 b 交换后的值),1 个命令按钮 Command1,其 Caption 属性值为

"交换",单击此按钮执行过程 Swap2 的代码。程序代码如下:

```
Public Sub swap2(ByRef x As Integer, ByRef y As Integer)   '形参定义为传地址方式
  Dim t As Integer
  t=x:x=y:y=t
End Sub
Private Sub Command1_Click()
  Dim a As Integer, b As Integer
  a=Val(Text1.Text)
  b=Val(Text2.Text)
  swap2 a,b                    '调用子过程 Swap2
  Text3.Text=a
  Text4.Text=b
End Sub
```

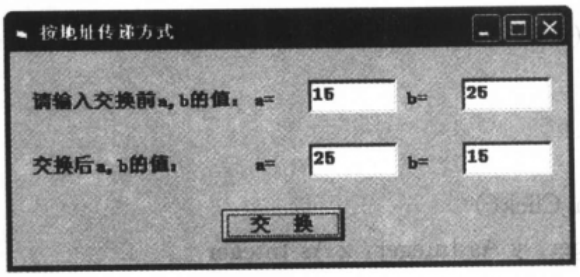


图 6-6 例 6-4 运行界面

分析:从输出结果可以看到,调用子过程 Swap2 可以实现交换变量的值,其原因是:过程 Swap2 采用地址传递方式,过程被调用时,形参 x 和 y 获得实参 a 和 b 的地址,即 x 和 a 使用同一存储单元,y 和 b 使用同一存储单元[如图 6-7(b)]。因此,在被调过程 Swap2 中,x 和 y 通过临时变量 t 实现交换后,实参 a 和 b 的值也同样被交换[如图 6-7(c)]。当调用结束返回主调过程后,形参 x 和 y 的临时内存单元将释放,实参单元 a 和 b 的值就是交换后的值[如图 6-7(d)]。地址传递的整体执行过程如图 6-7 所示。

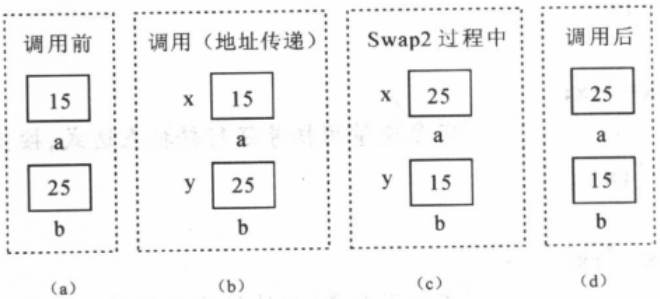


图 6-7 地址传递执行过程

说明:

(1) 在定义过程时,一般要求说明形参变量的数据类型,若形参被说明成默认类型,则

此时形参为 Variant 数据类型，由调用时实参的数据类型来确定，这样程序的执行效率低，且容易出错。

(2) 当按地址传递时，实参与形参的数据类型必须相同，否则就会出错。当按值传递时，实参数据类型与形参类型不同，系统将实参的数据类型转换为形参的数据类型，然后再传递给形参，如果实参的数据类型不能转换为形参的数据类型，则会出错。

(3) 在过程调用时，如果实参是常量(包括系统常量、用 Const 自定义的符号常量)或表达式，无论在定义时使用值传递还是地址传递，此时都是按值传递方式将常量或表达式计算的值传递给形参变量。

如果形参定义是按地址传递方式，但调用时想使实参变量按值方式传递，可以把实参变量加上括号，将其转换成表达式即可。

【例 6-5】 下面程序窗体的单击事件中，4 次调用过程 Sabc，每次使用不同的实参数，试分析程序的运行结果。

解题思路：程序运行时，单击窗体便可得到运行结果，结果直接显示在窗体界面中。程序代码如下：

```
Private Sub Sabc(a As Integer)
  a=a+5
  Print "a=";a,
End Sub
Private Sub Form_Click()
  Dim x As Integer, y As Integer, z As Integer
  x=10:y=20
  Print "1:…x=";x;
  Sabc x+y                    '实参是用表达式,按值传方式传递
  Print "…x=";x
  Print
  z=x+y
  Print "2:…z=";z;
  Sabc z                      '实参用变量,按地址方式传递
  Print "…z=";z
  Print
  Print "3:…x=";x;
  Sabc (x)                    '实参变量用括号强行转换表达式,按值传方式传递
  Print "…x=";x
  Print
  Print "4:…x=";x;
  Sabc x                      '实参用变量,按地址方式传递
  Print "…x=";x
End Sub
```

程序的运行结果为：

1:…x=10 a=35 …x=10
2:…z=30 a=35 …z=35
3:…x=10 a=15 …x=10
4:…x=10 a=15 …x=15

(4) 解决一个问题既可以使用自定义过程,也可以使用函数过程。如果是需要求得一个值,一般情况使用函数过程,如不是为了求一个值,而是完成一些操作,或需要返回多个值,则使用自定义。

【例6-6】 计算 n! /[m! ×(n-m)!],分别用 Function 过程和 Sub 过程实现。

解题思路:程序运行界面如图6-8所示。在窗体上建立3个标签,4个文本框(上面2个从左到右分别为 Text1, Text2,用来输入 m 和 n 的值;下面2个从左到右分别为 Text3, Text4,用来显示使用 Function 过程和 Sub 过程求解后的值),2个命令按钮(Command1,其 Caption 属性值为"用 Function 过程求解",单击此按钮执行用 Function 过程求解的代码;Command2,其 Caption 属性值为"用 Sub 过程求解",单击此按钮执行用 Sub 过程求解的代码)。程序代码如下:

```
Dim m As Integer,n As Integer                          '在窗体的通用部分声明
Private Function jc1(ByVal x As Integer) As Double
  Dim i As Integer
  Dim result As Double
  result=1
  For i=1 To x
    result=result * i
  Next i
  jc1 = result
End Function
Private Sub Command1_Click()
  Dim doufact As Double
  m=Val(Text1.Text)
  n=Val(Text2.Text)
  doufact=jc1(n)/(jc1(m) * jc1(n -m))                  '通过函数调用得到阶乘值
  Text3 = Str(doufact)
End Sub
Private Sub jc2(ByVal x As Integer, ByRef result As Double)
  Dim i As Integer
  result=1
  For i=1 To x
    result=resul t * i
  Next i
End Sub
Private Sub Command2_Click()
```

```
    Dim r1 As Double, r2 As Double, r3 As Double
    m = Val(Text1.Text)
    n = Val(Text2.Text)
    Call jc2(n, r1)              '通过引用传递参数 r1 带回自定义过程的阶乘值
    Call jc2(m, r2)
    Call jc2(n - m, r3)
    Text4.Text = Str(r1/(r2 * r3))
End Sub
```

从上面的例题可见，一个可以用函数实现的过程，也可以使用 Sub 过程来完成，但必须通过传地址方式的形参变量来带回函数值，如上面的 r1，r2 和 r3。

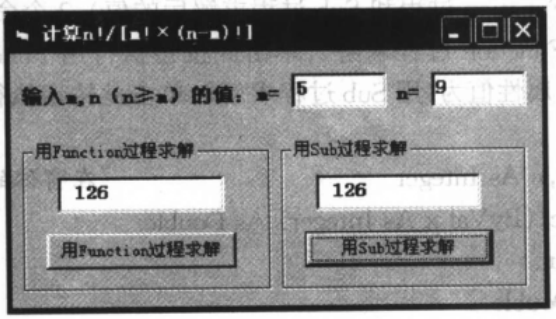


图 6-8 例 6-6 运行界面

6.4.3 数组参数的传递

Visual Basic 允许把数组作为实参传送到过程中，数组一般通过地址传递的方式传送。在传送数组时，除遵守参数传送的一般规则外，还应注意以下几点：

(1) 为了把一个数组的全部元素传送给一个过程，应将数组名写入形参表中，并略去数组的上下界，但括号不能省略。

例如：

```
Private Sub array(a() As single)
  ...
End Sub
```

其中形参“a()”即为数组。

(2) 被调过程可通过 Lbound 和 Ubound 函数确定形参数组的上、下界。

(3) 当用数组作形参时，对应的实参必须也是数组，且类型一致。

(4) 实参和形参结合是按地址传递，即形参数组和实参数组共用一段内存单元。

例如，定义实参数组 b(1 to 8)，给它们赋值，调用 array()过程的形式如下：

```
array  b()  或  Call array (b())
```

实参数组后面的括号可以省略，但为便于阅读，建议一般不要省略。

调用时形参数组 a 和实参数组 b 虚实结合，共用一段内存单元，如图 6-9 所示。因此在 array ()过程中改变数组 a 的各元素值，也就相当于改变实参数组 b 中对应的元素值，当调用结束时，形参数组 a 将无意义。

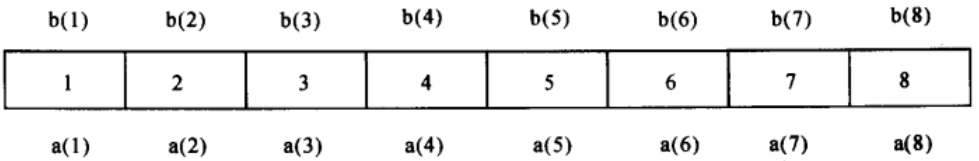


图 6-9 参数为数组时实参与形参内存单元对应图

【例 6-7】 输入若干个(不超过 100)学生的成绩,求出平均分、最高分及最低分。

解题思路:采用 InputBox 函数输入成绩,计算结果直接输出到窗体上,程序代码如下:

```
Sub caljc(k As Integer, darray() As Integer,s As Long, m As Integer, n As Integer)
  Dim i As Integer
  s=darray(1):m=darray(1):n=darray(1)
  If k=1 Then Exit Sub
  For i=2 To k
    s=s+darray(i)
    If m<darray(i) Then m=darray(i)
    If n>darray(i) Then n=darray(i)
  Next i
End Sub
Private Sub Form_Load()
  Dim jc(100) As Integer, x As Integer
  Dim n As Integer, sum As Long
  Dim max As Integer,min As Integer
  n=0
  Do While True
    x=Val(InputBox("请输入第" & n + 1 & "个学生的成绩(-1结束)"))
    If x=-1 Then Exit Do
    n=n+1
    jc(n)=x
  Loop
  If n>0 Then
    Call caljc(n, jc(), sum, max, min)
  Else
    End
  End If
    Show
    Print "平均分:";Format(sum/n,"###.0")
    Print "最高分:";max
    Print "最低分:";min
End Sub
```

程序输入 10 个学生的成绩:67,78,90,85,56,34,87,67,76,72

运行结果为:平均分:71.2

最高分:90

最低分:34

6.4.4 对象参数的传递

除了用一般的变量、数组作为形参外,Visual Basic 还允许用对象作为参数,如窗体和控件。这样做可以简化程序设计,提高效率。用对象作为参数与用一般变量作为参数的过程没有什么区别,具体格式为:

```
Sub 过程名(对象形参表)
  <语句组>
  [Exit Sub]
  <语句组>
End Sub
```

"对象形参表"中的形参类型通常为 Control(控件对象)或 Form(窗体对象)。

注意:在调用含有对象的过程时,对象只能按地址方式传递。

1.窗体参数

当多个窗体对象的属性或完成的功能相同时,可以用窗体作为参数编写一个自定义过程。

【例 6-8】 设计一个多窗体程序,共 3 个窗体,分别为 form1、form2 和 form3。form2 和 form3 窗体的大小和位置都是相同的,form1 为启动窗体,在该窗体中编写一个自定义过程 showform(frm as form,c as string),同时 form1 窗体界面放置 2 个命令按钮 Command1 和 Command2,在二者的单击事件过程中,分别用 form2、form3 作为参数调用 showform 过程,实现对不同窗体的显示。

解题思路:在 form1 中单击 Command1 按钮,显示 form2 窗体,并且在该窗体中显示"欢迎进入窗体 2"文本;在 form1 中单击 Command2 按钮,显示 form3 窗体,并且在该窗体中显示"欢迎进入窗体 3"文本。程序代码如下:

```
Private Sub showform(frm As Form, c As String)
  frm.Show
  frm.Left=2000
  frm.Top=3000
  frm.Width=5000
  frm.Height=3000
  frm.Print c
End Sub
Private Sub command1_click()
  Call showform(Form2,"欢迎进入窗体 2")
  End Sub
  Private Sub command2_click()
  Call showform(Form3,"欢迎进入窗体 3")
End Sub
```

2.控件参数

控件参数和窗体参数一样,都可作为自定义过程的参数,即在一个自定义过程中设置相同性质所需要的属性,然后用不同的控件调用此过程。

【例6-9】 控件参数使用示例。

解题思路:程序运行界面如图6-10所示。在窗体上建立1个标签(Label1),其Caption属性值为"单击"移动标签"按钮进行移动",1个文本框(Text1),其Text属性值为"单击"移动文本框"按钮进行移动",2个命令按钮(Command1,其Caption属性值为"移动文本框",单击此按钮将以文本框控件为参数调用movecontrol过程,实现界面中文本框的移动;Command2,其Caption属性值为"移动标签",单击此按钮将以标签控件为参数调用movecontrol过程,实现界面中标签的移动)。程序代码如下:

```
Private Sub movecontrol(ctr As Control,x As Single, y As Single)
  If ctr.Top+ctr.Height<Form1.Height Then
    ctr.Move ctr.Left+x, ctr.Top+y
  Else
    ctr.Top=0
    ctr.Left=0
  End If
End Sub
Private Sub Command1_Click()
  Dim x As Single, y As Single
  x=200
  y=400
  Call movecontrol(Text1,x,y)
End Sub
Private Sub Command2_Click()
  Dim x As Single,y As Single
  x=200
  y=300
  Call movecontrol(Label1,x,y)
End Sub
```

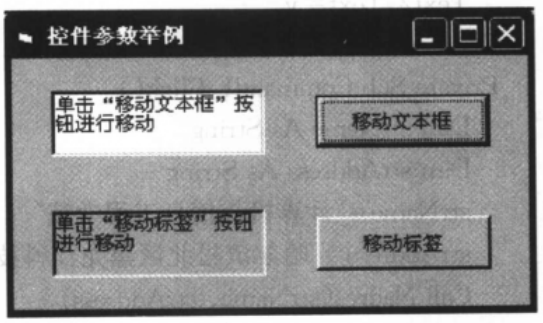


图6-10　例6-9运行界面

6.5　可选参数和可变参数

在过程调用时,要求实参个数、类型与形参的个数、类型一一对应,为了增加参数传递的灵活性,Visual Basic还允许在参数传递时使用可选参数和可变参数。在调用一个过程时,可以向过程传送可选的参数或任意数量和类型的参数。

6.5.1　可选参数

要指定某个形参为可选参数,在形参变量前加入Optional关键字即可。如果一个过程某个形参为可选参数,则在调用此过程时可以不提供对应于这个形参的实参,若一个过程有

多个形参，当它的一个形参被设定为可选参数，则这个形参后面的其他参数也必须是可选的，并且要用 Optional 关键字声明。

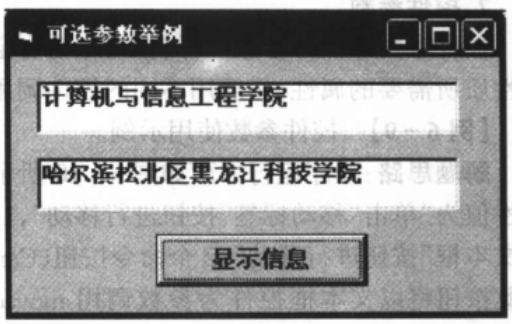


图 6－11　例 6－10 运行界面

【例 6－10】 可选参数举例。

解题思路：程序运行界面如图 6－11 所示。在窗体上建立 2 个文本框（Text1 和 Text2，Text1 在程序运行时用于显示单位名称，Text2 在程序运行时用于显示单位地址），1 个命令按钮（Command1，其 Caption 属性值为“显示信息”，程序运行时，单击此按钮将在上面 2 个文本框中显示单位信息和地址信息）。程序代码如下：

```
Private Sub Nadres(x As String, Optional y As String)
  Text1.Text = x
  Text2.Text = y
End Sub
Private Sub Command1_Click()
  Dim strName As String
  Dim strAddress As String
  strName = "计算机与信息工程学院"
  strAddress = "哈尔滨松北区黑龙江科技学院"
  Call Nadres(strName, strAddress)          '提供了两个参数
End Sub
```

说明：

（1）可选参数类型一般为 Variant 型。在过程中可以使用 Visual Basic 的内部函数 IsMissing 来测试过程被调用时，是否向可选参数传递了实参值。IsMissing 具体格式为：

IsMissing(可选参数名)

如果在调用过程时，没有向可选参数传递实参，则 IsMissing 返回值为 True，否则返回 False。

注意：若可选参数是 Variant 以外的类型，则 IsMissing 总是返回 False，不能起到检测的作用。

例如，将例 6－10 的 Nadres 过程修改如下：

```
Private Sub Nadres(x As String, Optional y As String)
  Text1.Text = x
  If IsMissing(y) Then
    Text2.Text = y
  Else
    Text2.Text = "信息与计算科学专业"
  End If
End Sub
```

则 text2 中将显示“信息与计算科学专业”文本。

如果将例 6－10 的 Nadres 过程修改如下：

```
Private Sub Nadres(x As String, Optional y As Variant)
  Text1.Text=x
  If IsMissing(y) Then
  Text2.Text="信息与计算科学专业"
  Else
  Text2.Text=y
  End If
End Sub
```

则 Text2 中将显示“哈尔滨松北区黑龙江科技学院”文本。

(2) 也可以给可选参数指定一个默认值。如果在调用过程时，没有向可选参数传递实参，则该可选参数将使用该默认值。

例如，将例 6－10 修改如下：

```
Private Sub Nadres(x As String,Optional y As String="哈尔滨松北区黑龙江科技学院")
  Text1.Text=x
  Text2.Text=y
End Sub
Private Sub Command1_Click()
  Dim strName As String
  strName="计算机与信息工程学院"
  Call Nadres(strName)              '只提供了 1 个参数。
End Sub
```

程序的运行界面如图 6－11 所示。

6.5.2 可变参数

可变参数即在参数传递时不要求实参和形参类型一致，可以传递任何类型的参数。可变参数过程通过 ParamArray 命令来定义。具体格式为：

```
Sub  过程名(ParamArray 数组名())
```

或

```
Function  函数名(ParamArray 数组名())[As 数据类型]
```

这里数组名是一个形参，只有名字和括号，没有上下界，是一个动态数组。数组的类型默认为 Variant，由于可变参数过程中的形参是 Variant 类型，因此可以把任何类型的实参传递给该过程，同时该过程可以接受任意个数的参数。

注意：关键字 ParamArray 只能用于参数表中的最后一个参数，而且不能和关键字 ByVal、ByRef 以及 Optional 一起使用。

【例 6－11】 建立一个求多个数乘积的过程。

解题思路：建立一个窗体，在窗体中放置 1 个命令按钮(Command1)，单击该命令按钮将得到程序的运行结果，该结果直接显示在窗体上。程序代码如下：

```
Private Sub Multiply(ParamArray a())
```

```
    Dim result As Single, x As Variant          '注意:x 必须定义为可变型
    result=1
    For Each x In a
      result=result * x
    Next x
    Print "result=";result;
End Sub
Private Sub command1_click()
    Multiply 3,5,8,7,4,9
    Multiply 3.2,2.3
End Sub
```

程序运行结果为:

result=30240 result=7.36

6.6 过程的嵌套和递归调用

6.6.1 过程的嵌套调用

Visual Basic 的过程定义都是互相平行和相对独立的,也就是说在定义过程时,一个过程内不能包含另一个过程。Visual Basic 虽然不能嵌套定义过程,但可以嵌套调用过程,也就是主程序可以调用子过程,而在子过程中又可以调用另外的子过程,这种程序结构称为过程的嵌套。

【例 6-12】 输入一个整数 n,求 1! +2! +…+n! 的值。

解题思路:程序运行界面如图 6-12 所示。在窗体上建立 2 个标签,2 个文本框(Text1 和 Text2,Text1 在程序运行时用于输入 n 的值,Text2 在程序运行时用于显示求得的 1! + 2! +…+n! 的值),1 个命令按钮(Command1,其 Caption 属性值为"计算",程序运行时,单击此按钮将在 text2 中显示求得的值)。

在程序中,采用 3 个过程嵌套调用的形式,在第一层的事件过程 Command1_Click 中,提示用户输入 n 的值,通过调用子程序 calculate 进行公式的计算和结果的输出,而在子程序的公式计算中,只进行累加的计算,1~n 之间每个数的阶乘计算又通过调用函数 fact 实现,把阶乘返回到子程序 calculate 中进行累加。程序代码如下:

```
Public Function fact(k)
    Dim j, t As Integer
    t=1
    For j=1 To k
      t=t * j
    Next j
    fact=t
End Function
Public Sub calculate(m)
```

```
    Dim i, s As Integer
    s=0
    For i=1 To m
      s=s+fact(i)
    Next i
    Text2.Text=s
  End Sub
  Private Sub Command1_click()
    Dim n As Integer
    n=Val(Text1.Text)
    Call calculate(n)
  End Sub
```

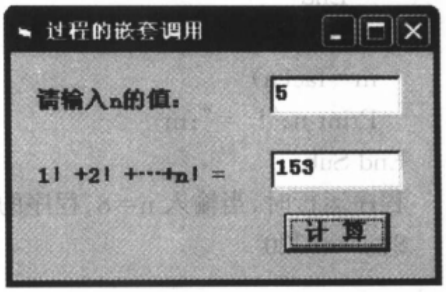


图6-12 例6-12运行界面

6.6.2 过程的递归调用

一个过程调用过程本身,就称为过程的递归调用。采用递归方法来解决问题时,必须符合以下两个条件:

(1) 可以把要解决的问题转化为一个新的问题,而这个新问题的解法仍与原来的解法相同。

(2) 有一个明确的结束递归的条件(终止条件),否则过程将永远"递归"下去。

【例6-13】 采用递归方法求n!(n>0)。

解题思路:求n!,可以采用递归的方法,即n! =n×(n-1)!,而(n-1)! =(n-1)×(n-2)!,…,1! =1,因此可用下列递归公式:

$$n! = \begin{cases} 1 & n=1 \\ n\times(n-1) & n>1 \end{cases}$$

此递归算法中,终止条件是n=1。

程序使用InputBox输入n的值为(1~15)之间的整数,计算结果直接显示在窗体上。

程序代码如下:

```
Private Function fac(n) As Double
  If n>1 Then
    fac=n * fac(n-1)                    '递归调用
  Else
    fac=1                               '=1时,结束递归
  End If
End Function
Private Sub Form_Load()
  Dim n As Integer,m As Double
  Show
  n=Val(InputBox("输入1~15之间的整数"))
  If n<1 Or n>15 Then
    MsgBox "错误数据",0,"检查数据"
```

```
    End
  End If
  m=fac(n)
  Print n;"! =";m
End Sub
```

程序运行时，当输入 n=8，程序的运行结果为：

8! =40320

6.7 鼠标与键盘事件过程

Visual Basic 应用程序能够响应多种鼠标事件和键盘事件。例如，窗体、图片框等控件都能检测鼠标指针的位置，并且可以判定其左、中、右键是否已经按下，还能响应鼠标按钮与键盘的 Shift、Ctrl 或 Alt 键的各种组合。利用键盘事件还可以编程响应多种键盘操作，也可以解释、处理 ASCII 字符。此外，Visual Basic 应用程序还可同时支持事件驱动的拖放功能。

6.7.1 鼠标事件过程

1. MouseMove、MouseDown 和 MouseUp 事件

大多数控件能够识别鼠标的 MouseMove、MouseDown 和 MouseUp 事件，通过响应这些鼠标事件，能在应用程序中对鼠标位置及状态的变化做出响应操作。

(1) MouseMove 每当鼠标指针移动到屏幕新位置时发生。

(2) MouseDown 按下任意鼠标键按钮时发生。

(3) MouseUp 释放任意鼠标键按钮时发生。

MouseMove、MouseDown、MouseUp 三个事件过程的语法格式为：

```
Sub Object_MouseMove(Button As Integer,Shift As Integer,X As Single,Y As Single)
Sub Object_MouseDown(Button As Integer,Shift As Integer,X As Single,Y As Single)
Sub Object_MouseUp(Button As Integer,Shift As Integer,X As Single,Y As Single)
```

说明：

(1) Object 是可选的一个对象表达式，可以是窗体对象和大多数可视控件。

(2) Button 是表示按下或松开鼠标某个按钮的参数值，当按下或松开鼠标的不同按钮，得到的值是不同的。如图 6-13 所示，Button 参数所对应的二进制数低 3 位 M、R、L 分别

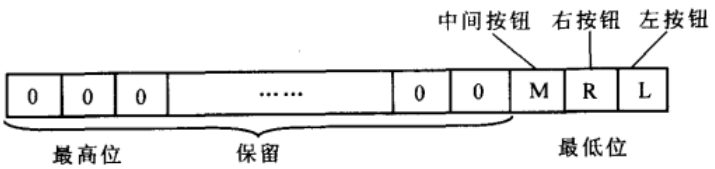


图 6-13 Button 参数的值

表示中间按钮、右按钮和左按钮的状态，相应二进制位为 0 时表示未按下对应按钮，相应二进制位为 1 时表示按下对应按钮。因此，当按下左键 Button 参数的值为 1(二进制为 001)；当按下右键 Button 参数的值为 2(二进制为 010)；当按下中间键 Button 参数的值为 4(二进制为 100)。

(3) Shift 参数表示在 Button 参数指定的按钮被按下或者被松开的情况下，键盘的 Shift、Ctrl 和 Alt 键的状态，以编写用于鼠标键盘组合操作的代码。如图 6－14 所示，Shift 参数所对应的二进制数低 3 位 A 、C、S 分别表示 Alt、Ctrl 和 Shift 键的状态，相应二进制位为 0 时表示未按下对应键，为 1 时表示按下对应键。因此，当按下 Shift 键参数的值为 1(二进制为 001)；当按下 Ctrl 键参数的值为 2(二进制为 010)；当按下 Alt 键参数的值为 4(二进制为 100)。

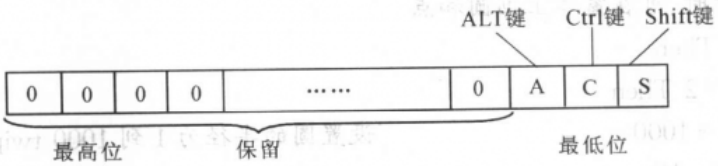


图 6－14 Shift 参数的值

注意：Button 参数和 Shift 参数可以表示重选的情况。例如，同时按下鼠标左右两键时，Button 参数的二进制值为 011；同时按下 Alt 键和 Ctrl 键时，Shift 参数的二进制值为 110。

【例 6－14】 利用鼠标的 MouseMove、MouseDown 和 MouseUp 事件实现在窗体上画图。

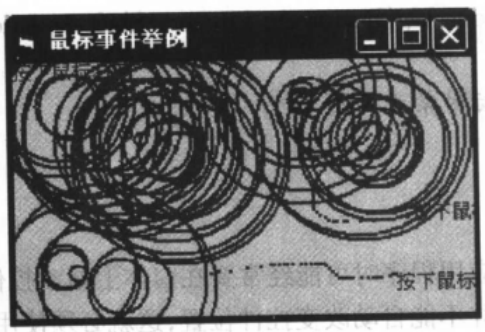


图 6－15 例 6－14 运行界面

解题思路：程序运行时，按下鼠标左键或右键，可以在窗体上画图，如图 6－15 所示，程序代码如下：

```
Dim paintnow As Boolean                    '在窗体通用部分声明
Dim trace As Integer
Private Sub Form_DblClick()                '双击窗体时清屏
  Cls
End Sub
Private Sub Form_Load()                    '设置画笔的宽度为 2，并设置画图的颜色
  DrawWidth=2
  ForeColor=RGB(0,255,0)
End Sub
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, _
Y As Single)
```

```
'当按下鼠标时允许画图,按下鼠标右键时在窗体显示"按下鼠标右键"的提示
  If Button=2 Then Print "按下鼠标右键"
  paintnow=True
End Sub
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, _
Y As Single)
'当移动鼠标时,可在窗体上画圆和点
If paintnow Then
  If Button=2 Then
    r=Rnd*1000                    '设置圆的半径为1到1000 twip的一个随机数
    Circle(X, Y),r
  Else
    PSet(X,Y)
  End If
End If
End Sub
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As _
Single)
  '在窗体上松开鼠标时禁止画图
  paintnow = False
End Sub
```

2.拖放操作

在设计 Visual Basic 应用程序时可能经常要在窗体上拖动控件,改变其位置。但在运行时拖动控件,通常情况下并不能自动改变控件位置,这就必须使用 Visual Basic 的拖放功能,通过编程才能实现在运行时拖动控件并改变其位置的操作。拖放可以分解为两个操作:把按下鼠标按钮并移动控件的操作称为拖动,把释放按钮的操作称为放下。

通常拖放操作要使用到表 6-1 中所列的拖放属性、事件和方法。

表 6-1 拖放属性、事件和方法

类别	名称	描述
属性	DragMode	启动自动拖动控件或手工拖动控件操作
	DragIcon	指定拖动控件时显示的图标
事件	DragDrop	识别何时将控件拖动到对象上
	DragOver	识别何时在控件上拖动对象
方法	Drag	启动或停止手工拖动

说明:除菜单、时钟、直线、形状控件以外的所有控件均支持 DragMode、DragIcon 属性和 Drag 方法。窗体识别 DragDrop 和 DragOver 事件,但不支持 DragMode、DragIcon 属性和

Drag 方法。

(1) DragMode 属性:用来返回或设置一个值,确定在拖放操作中所用的是手动还是自动拖动方式。其使用格式为:

Object. DragMode[= number]

其中:Object 为对象表达式,其值是支持 DragMode 属性的一个对象。number 为整数,指定拖动方式,其取值为 0 或 1。

说明:

① 当 DragMode 属性取默认值 0(内部常数是 VbManual)时,即启动人工拖放模式。这就需要在源控件的 MouseDown 事件中,用源控件的 Drag 方法来启动拖放操作。

② 当 DragMode 属性设置为 1(内部常数是 VbAutomatic)时,即启动自动拖放模式。在此模式下,当用户在源对象上按下鼠标左键同时拖动鼠标时,对象的图标便随指针移动到目标对象上;当释放鼠标时,在目标对象上产生 DragDrop 事件。但是如果没有对目标对象的 DragDrop 事件编程序,对象本身不会移动到新位置或被添加到目标对象中。因此,要实现真正拖放就要在目标对象的 DragDrop 事件中编写相应的程序。在源对象被拖到目标对象的过程中,如果经过其他对象,则在这些对象上会产生 DragOver 事件。当然在目标对象上也会产生 DragOver 事件,它发生在 DragDrop 事件之前。

③ 当 DragMode 属性设置为 1(自动拖放模式)时,控件不能正常响应 Click、DbClick 和 MouseDown 事件;当拖动控件时,该控件不能识别用户发出的其他鼠标或键盘事件(KeyDown、KeyPress 或 KeyUp,MouseDown、MouseMove 或 MouseUp)。

(2) DragIcon 属性:用来返回或设置图标,在拖动操作中作为指针显示,默认情况下是将源对象的灰色轮廓作为拖动图标。其使用格式为:

Object. DragIcon[= icon]

其中:Object 为对象表达式,其值是支持 DragMode 属性的一个对象。icon 为任何返回图标的引用,例如引用窗体图标(如 Form1. Icon),引用另外控件的 DragIcon 属性(如 Text1. DragIcon),或是 LoadPicture 函数装载图标文件(* . ico, * . cur)。

说明:

① 设置 DragIcon 属性的最简单的方法就是使用属性窗口。选定 DragIcon 属性后单击"属性"按钮,再从"加载图标"对话框中选择包含图形图像的文件。

② 运行时,DragIcon 属性可以设置为任何对象的 DragIcon、Icon 属性、Picture 属性,或者可以用 LoadPicture 函数将一个. ico 文件返回的图标给它赋值。

例如:

```
Set Image1. DragIcon = Image2. DragIcon
Set Image1. DragIcon = LoadPicture("d: \ ttt. ico")
```

(3) Drag 方法:当一个控件的 DragMode 属性取值为 0(人工拖放模式)时,使用 Drag 方法控制拖放的启动和停止。其使用格式为:

Object. Drag[Action]

表 6-2 列出了 Action 的取值与控制拖动操作的对应关系。

例如:在窗体中放置一个文本框(Text1),采用自动方式,把 DragMode 设置为 1;或者采用手工方式,把 DragMode 设置为 0。但在文本框的鼠标按下事件中编写:

Text1.Drag 1

表 6-2 Action 的取值与控制拖动操作的对应关系

Action 整数值	符号常量	意义
0	VBCancel	取消拖动操作
1	VBBeginDrag	启动拖动操作
2	VBEndDrag	结束拖动操作

这两种方法都可以达到拖动文本框的目的。

(4) DragDrop 事件:在一个完整的拖放动作(即将一个控件拖动到一个对象上,并释放鼠标按钮)完成时,或使用 Drag 方法,并将其 Action 参数设置为 2 时,该事件发生,用来实现在拖放操作完成时要进行的处理。其使用格式为:

Private Sub Object_DragDrop([Index As Integer,]Source As Control,x As Single, _
y As Single)

其中各参数的含义如下:

Object:对象表达式,其值是支持 DragMode 属性的一个对象。

Index:为一个整型数,可选参数,用来惟一地标识一个在控件数组中的控件。

Source:正在被拖动的控件。可在事件过程中用此参数使用或设置控件属性以及调用控件的方法。

例如:

Source.Visible = False

x,y:用来指定当前鼠标指针在目标窗体或控件中水平(x)和垂直(y)的位置。这些坐标值通常用目标坐标系统来表示,该坐标系是通过 ScaleHeight、ScaleWidth、ScaleLeft 和 ScaleTop 属性而设置的。

说明:

① DragDrop 事件过程用来控制在一个拖动操作完成时将会发生的情况。例如,可将源控件移到一个新的位置或将一个文件从一个位置复制到另一个位置。

② 当 Source 参数中可能使用多个控件时,需要按下列方法进行处理:

应使用 TypeOf 关键字和 If 语句一起确定 Source 表示控件的类型;

应使用该控件的 Tag 属性标识一个控件,然后使用 DragDrop 事件过程。

③ 应首先使用 DragMode 属性和 Drag 方法指定开始拖动的操作。一旦开始拖动,可使用 DragOver 事件过程处理位于 DragDrop 事件前面的事件。

(5) DragOver 事件:当源对象被拖动到目标对象上时,在目标对象上会触发 DragOver 事件,该事件先于 DragDrop 事件被执行。DragOver 事件可用来设置被拖动对象放在目标对象上之前的状态,如加亮目标,显示一个特定的拖动指针等。其使用格式为:

Private Sub Object_DragOver([Index As Integer,]Source As Control,x As Single, _
y As Single,State As Integer)

其中:Object、Index、Source、x 和 y 参数含义同 DragDrop 事件。State 参数是一个 0、1 或 2 的整数,0 表示进入,即源对象正进入目标对象内;1 表示离开,即源对象正离开目标对

象;2 表示跨越,即源对象正在目标对象范围内移动位置。

【例 6-15】 在窗体上放置1 个Image 控件、1 个 Picture 控件,1 个 Label 控件,设计一个如图 6-16 所示的应用程序,实现对象的拖放操作。

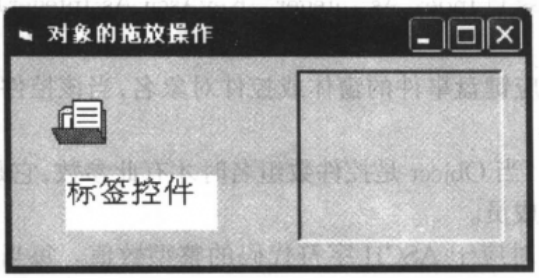


图 6-16 例 6-15 运行界面

具体操作如下:将 Image 控件 Image1 作为源控件,给 Image1 控件加载代表"打开状态形文件夹"的图标文件(当然可在 Image1 属性中加载任何其他图标文件),把它的 DragMode 属性设置为"1 - Automatic",DragIcon 属性设置为"处于关闭状态形的文件夹"的图标文件(当然也可加载任何其他图标文件)。当把 Image1 拖到 Picture 控件 Picture1 中,放开鼠标,源图标将消失,"打开状态形文件夹"的图标将出现在 Picture1 中。当源控件 Image1 被拖动经过标签控件 Label1 时,则取消拖放操作。

程序代码如下:

```
Private Sub Picture1_DragDrop(Source As Control, X As Single, Y As Single)
  If TypeOf Source Is Image Then
    Picture1.Picture = Image1.Picture
    Source.Visible = False
  End If
End Sub
Private Sub Label1_DragOver(Source As Control, X As Single, Y As Single, _
State As Integer)
  Source.Drag vbCancel                    '取消拖放操作
End Sub
```

6.7.2 键盘事件过程

虽然 Windows 应用程序常常使用鼠标操作,但有时也需要使用键盘操作,尤其是对于接受文本输入的控件,例如文本框 TextBox,若需要控制文本框中输入的内容,处理 ASCII 字符等,就需要对键盘事件编程。

在 Visual Basic 中,提供了 KeyPress、KeyDown、KeyUp 三种键盘事件,窗体和接受键盘输入的控件都识别这三种事件。

(1) KeyPress 按下对应某 ASCII 字符的键。

(2) KeyDown 按下键盘的任意键。

(3) KeyUp 释放键盘的任意键,只有获得焦点的对象才能够接受该键盘事件。

1. KeyPress 事件

在按下与 ASCII 字符对应的键时,将触发 KeyPress 事件。ASCII 字符集不仅代表标准

键盘的字母、数字和标点符号，而且也能代表大多数控制键。但是，KeyPress 事件只能识别 Enter、Tab 和 BackSpace 键，不能够检测其他功能键、编辑键和定位键。

KeyPress 事件过程的语法格式为：

Sub Object_KeyPress ([Index As Integer,]KeyAscii As Integer)

其中：

Object 是指可以响应键盘事件的窗体或控件对象名，当该控件获得焦点时才能响应键盘事件。

Index 是一整型数。当 Object 是控件数组名时才有此参数，它是控件数组元素的下标，指定数组中的一个控件成员。

KeyAscii 参数返回对应于 ASCII 字符代码的整型数值。每当键盘上的一个有 ASCII 码的键被按下时，该键的 ASCII 码就被存储到变量 KeyAscii 中，引发 KeyPress 事件。

例如，如果希望将文本框中的所有字符都强制转换为大写字符，则可在输入时使用此事件转换大小写：

```
Private Sub Text1_KeyPress (KeyAscii As Integer)
  KeyAscii = Asc(Ucase(Chr(KeyAscii)))
End Sub
```

上述过程 Chr 将 ASCII 字符代码转换成对应的字符，然后用 Ucase 将字符转换为大写，并用 Asc 将结果转换回字符代码。

实际应用中，可以通过 KeyPress 事件识别一个键。

例如，下述事件过程中使用 KeyPress 事件检测用户是否正在按 BackSpace 键：

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
  If KeyAscii = 8 Then
    MsgBox "你正在按 BackSpace 键"
  End If
End Sub
```

也可用 Visual Basic 的键代码常数代替字符代码。示例中 BackSpace 键的 ASCII 值为 8，其常数值为 vbKeyBack，因而“If KeyAscii = 8 Then”也可写为“If KeyAscii = vbKeyBack Then”。更多的键代码常数可参见“视图”菜单→“对象浏览器”命令→“KeyCodeConstants”项。

【例 6-16】 编写一程序，实现在一个文本框(Text1)中限定只能输入数字、字母和下划线，且只能响应 BackSpace 键。

解题思路：程序中根据限定的各个字符的 ASCII 码进行操作，数字的 ASCII 码范围为 48～57；字母的 ASCII 码范围为 65～90 和 97～122 两段；下划线的 ASCII 码为 95；BackSpace 键的 ASCII 码为 8。当按其他键时，让 KeyAscii = 0，即不受该操作限制。程序代码如下：

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
  If KeyAscii<>8 Then
    If 48>KeyAscii Or KeyAscii>122 Then
      MsgBox "只能输入数字、字母或下划线"
```

```
        KeyAscii = 0
      Else
        Select Case KeyAscii
          Case 58 To 64
            MsgBox "只能输入数字、字母或下划线"
            KeyAscii = 0
          Case 91 To 94
            MsgBox "只能输入数字、字母或下划线"
            KeyAscii = 0
          Case 96
            MsgBox "只能输入数字、字母或下划线"
            KeyAscii = 0
        End Select
      End If
    End If
  End Sub
```

2. KeyDown 和 KeyUp 事件

当一个对象具有焦点时，按下（KeyDown）或松开（KeyUp）一个键时将触发 KeyDown 或 KeyUp 事件。它们报告键盘本身准确的物理状态：按下键（KeyDown）及松开键（KeyUp）。与此成对照的 KeyPress 事件并不直接地报告键盘状态，它只能提供键所代表的字符而不能识别键的按下或松开状态。KeyDown 或 KeyUp 事件能够检测其他功能键、编辑键和定位键。

KeyUp 和 KeyDown 事件过程的语法格式为：

```
Sub Object_KeyDown([Index As Integer,]KeyCode As Integer, Shift As Integer)
Sub Object_KeyUp([Index As Integer,]KeyCode As Integer, Shift As Integer)
```

说明：

(1) "Object"和"[Index As Integer,]"含义同 KeyPress 事件。

(2) "Keycode" 参数表示按键的键盘码，即按键的物理位置。

此时"A"与"a"作为同一个键返回，即具有相同的 Keycode 值。也就是说，大小写字母使用同一键，它们的 Keycode 值相同，为大写字母的 ASCII 码。上档键字符和下档键字符也是使用同一键，它们的 Keycode 值也是相同的，为下档字符的 ASCII 码。注意，键盘上的"1"和数字小键盘上的"1"被作为不同的键返回，尽管它们生成相同的字符，但它们的 Keycode 值却是不同的。表 6-3 列出了部分字符的 KeyUp 或 KeyDown 事件的 Keycode 值和对应的 KeyPress 事件的 KeyAscii 值。

Keycode 参数也是通过 ASCII 值或键代码常数来识别键的。字母键的 Keycode 参数与此字母的大写字符的 ASCII 值相同。所以"B" 和"b" 的 Keycode 都是由 Asc("B") 返回的数值。在下例中用 KeyDown 事件判断是否按下"B"或"b" 键：

表 6-3 部分字符的 Keycode 值和 KeyAscii 值对照表

键(字符)	Keycode 值	KeyAscii 值
"A"	&H41	&H41
"a"	&H41	&H61
"!"	&H31	&H21
"1"(大键盘上)	&H31	&H31
"1"(数字键盘上)	&H61	&H31
HOME 键	&H24	&H24
F10 键	&H79	无

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
  If KeyCode = Asc("B") Then
    MsgBox "You pressed the B or b key."
  End If
End Sub
```

KeyDown 或 KeyUp 事件可识别标准键盘上的大多数的控制键。其中包括功能键(F1～F16)、编辑键(HOME、PAGEUP、DELETE 等)、定位键(→、←、↑和↓)和数字小键盘上的键。可以通过键代码常数或相应的 ASCII 值检测这些键。例如,检测按下的键是否为 HOME 的代码如下:

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
  If KeyCode = vbKeyHome Then
    MsgBox "You pressed the HOME key."
  End If
End Sub
```

关于字符代码或键代码常数,可通过 MSDN 的"Visual Basic 文档"→"参考"→"其他信息"→"字符集"中的"字符集(0～127)"和"字符集(128～255)"得到它们的完整列表。也可通过"视图"菜单→"对象浏览器"命令→"KeyCodeConstants"项获得基本信息。

(3) "Shift" 参数表示是在该事件发生时响应 Shift、Ctrl 和 Alt 键的状态,它是一个整数。其含义与前面 MouseMove、MouseDown、MouseUp 事件中的 Shift 参数完全相同。

例如,用 Shift 参数判断是否按下字母的大写形式(按下 Shift + b 键):

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
  If KeyCode = Asc("B") And Shift = 1 Then
    MsgBox "You pressed the B key."
  End If
End Sub
```

数字与标点符号键的 KeyCode 值与键上数字的 ASCII 字符代码相同。因此"1"和"!"的 KeyCode 都是由 Asc("1") 返回的数值。此时为了检测"!",需要使用 Shift 参数。代码如下:

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
  If KeyCode = Asc("1") And Shift = 1 Then
    MsgBox "You pressed the ! key."
  End If
End Sub
```

【例 6-17】 编写一程序,当按下键盘上的某个键时,输出该键的 KeyCode 码。

解题思路:

(1) 首先建立一个 KeyDown 事件过程,用它的参数 KeyCode 实现题目要求。

(2) 因为 KeyCode 中保存的是一个数值,可调用 Chr $ 函数将其转化为相应的字符,然后输出,因为 ASCII 经常采用十六进制,所以可用 Hex $ 函数将 KeyCode 转化成十六进制数。

(3) 输出时,每输出一定数量后,就要求从下一行开始输出。可定义一个变量 i,控制每行输出的数量,每发生一次 KeyDown 事件,i 加上 1 后再赋给 i,当 i 的值累加到 10 的倍数时,下一次将从新的一行开始输出。当按下回车键时,下一次也将从新的一行开始输出。

程序运行界面如图 6-17 所示,结果直接显示在窗体上,程序代码如下:

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
  Static i
  i = i + 1
  If i Mod 10 = 0 Then
    Print Chr $ (KeyCode);"--";Hex $ (KeyCode);" ";
    Print:Print
  ElseIf KeyCode = 13 Then
    i = 0
    Print:Print:Print
  Else
    Print Chr $ (KeyCode);"--";Hex $ (KeyCode);" ";
  End If
End Sub
```

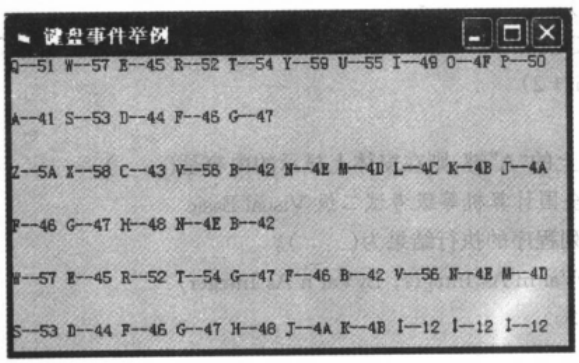


图 6-17 例 6-17 运行界面

习题六

1. 简答题

(1) 简述过程的定义及其分类。

(2) 试述子程序过程、函数过程和事件过程的区别。

(3) 在向过程传递参数时,有按值传递和按地址传递两种方式,它们有什么不同?

(4) 简述可选参数和可变参数的含义和适用场合。

(5) 简述鼠标事件 MouseMove、MouseDown 和 MouseUp 的基本含义和使用方法。

(6) 简述键盘事件 KeyPress、KeyDown、KeyUp 的基本含义和使用方法。

2. 填空题

(1) 单击一次命令按钮后,下列程序代码的执行结果是(　　)。

```
Public Function p(n As Integer)
  Static sum
  For i=q To n
    sum=sum+i
  Next i
  p=sum
End Function
Private Sub Command1_Click()
  s=p(1)+p(2)+p(3)+p(4)
  Print s;
End Sub
```

(2) 在鼠标事件中(如 MouseMove 事件),Shift 参数为 1 表示在操作鼠标的同时也按下键盘上的 Shift 键,Shitf 参数为 2 表示同时也按下键盘上的 Ctrl 键,那么 Shift 参数为 3 时,表示同时按下键盘上的(　　)键。

(3) 把窗体的 KeyPreview 属性设置为 True,然后编写如下事件过程:

```
Private Sub Form_KeyPress(KeyAscii As Integer)
  Dim ch As String
  ch=Chr(KeyAscii)
  KeyAscii=Asc(UCase(ch))
  Print Chr(KeyAscii+2)
End Sub
```

程序运行后,按键盘上的"A"键,则在窗体上显示的内容是(　　)。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(4) 单击窗体时,下列程序的执行结果为(　　)。

```
Private Sub value(ByVal m As Integer, ByVal n As Integer)
  m=m*2:n=n-5
  Print m;n
End Sub
Private Sub Form_Click()
  Dim x As Integer, y As Integer
```

```
  x=10:y=15
  Call value(x,y)
  Print x;y
End Sub
```

(5) 在窗体上放置1个名称为Command1的命令按钮和1个名称为Text1的文本框,然后编写如下程序:

```
Sub P1(ByVal a As Integer,ByVal b As Integer,c As Integer)
  c=a+b
End Sub
Private Sub Command1_Click()
  Dim x, y, z As Integer
  x=5:y=7:z=0
  Text1.text =""
  Call P1(x,y,z)
  Text1.text=Str(z)
End Sub
```

程序运行后,如果单击命令按钮,则在文本框中显示的内容为(　　)。

(6) 在窗体上画一个名称为Comamnd1的命令按钮,然后编写如下通用过程和命令按钮的事件过程:

```
Private Function fun(ByVal m As Integer)
  If m Mod 2=0 Then
    fun=2
  Else
    fun=1
  End If
End Function
Private Sub Command1_Click()
  Dim i As Integer,s As Integer
  s=0
  For i=1 To 5
    s=s+fun(i)
  Next
  Print s
End Sub
```

程序运行后,单击命令按钮,则窗体上显示的是(　　)。

【出处】2005年4月全国计算机等级考试二级Visual Basic

(7) 以下程序运行后,单击窗体时显示的结果为(　　)。

```
Sub mysub(x,t)
  t=0
  For k=1 To x
    t=t+k
  Next k
End Sub
Private Sub Form_Click()
```

```
  Dim b As Integer,y As Integer
  Call mysub(3,b)
  y=b
  Call mysub(4,b)
  Print y+b
End Sub
```

(8) 以下程序运行后,单击窗体时显示的结果为(　　)。

```
Function fn1(x) As String
  k=Len(x)
  fn1=Mid(x,2,k-2)
End Function
Private Sub Form_Click()
  Dim a As String,b As String,s As String
  a="ABCDEFG":b="12345"
  s=fn1(a)+fn1(b)
  Print fn1(fn1(fn1(s)))
End Sub
```

3. 选择题

(1) 下面的过程定义语句中,合法的是(　　)。

A. Sub Proc1(Byvaln())　　B. Sub Proc1(n) As Integer

C. Function Proc1(Proc1)　　D. Function Proc1(Byval n)

(2) Sub 过程与 Function 过程最根本的区别是(　　)。

A. Sub 过程可以使用 Call 语句或直接使用过程名调用,而 Function 过程不可以

B. Function 过程可以有参数,Sub 过程不可以

C. 两种过程参数的传递方式不同

D. Sub 过程不能返回值,而 Function 过程能返回值

(3) 当程序运行时,在窗体上单击鼠标,以下(　　)事件是窗体不会接收的。

A. MouseDown　　B. MouseUp　　C. Load　　D. Click

(4) 单击命令按钮时,下列程序代码的执行结果为(　　)。

```
Public Sub proc1(n As Integer,ByVal m As Integer)
  n=n Mod 10
  m=m Mod 10
End Sub
Private Sub Command1_Click()
  Dim x As Integer, y As Integer
  x=12:y=34
  Call proc1(x,y)
  Print x;y
End Sub
```

A. 12　34　　B. 2　34　　C. 2　3　　D. 12　3

(5) 单击窗体时,下列程序代码的执行结果为(　　)。

```
Private Sub proc1(x As Integer,y As Integer,z As Integer)
  x=3*z:y=2*z:z=x+y
```

```
End Sub
Private Sub Form_Click()
  Dim x As Integer, y As Integer,z As Integer
  x=1:y=2:z=3
  Call proc1(x,x,z)
  Print x;x;z
  Call proc1(x,y,y)
  Print x;y;y
End Sub
```

A. 6 6 12　　B. 9 6 15　　C. 9 6 15　　D. 9 10 10
6 10 10　　6 10 10　　9 10 15　　6 4 10

(6) 单击命令按钮时,下列程序代码的执行结果为(　　)。

```
Public Function myfunc(m As Integer,n As Integer) As Integer
  Do While m<>n
    Do While m>n:m=m-n:Loop
    Do While m<n:n=n-m:Loop
  Loop
  myfunc=m
End Function
Private Sub Command1_Click()
  Print myfunc(24,18)
End Sub
```

A. 2　　B. 4　　C. 6　　D. 8

(7) 在窗体上画一个名称为 Command1 的命令按钮,然后编写如下程序:

```
Dim SW As Boolean
Function func(X As Integer) As Integer
  If X<20 Then
    Y=X
  Else
    Y=20+X
  End If
  func=Y
End Function
Private Sub Command1_Click()
  Dim intNum As Integer
  intNum = InputBox("")
  If SW Then
    Print func(intNum)
  End If
End Sub
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
  SW=False
End Sub
```

```
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
  SW = True
End Sub
```

程序运行后，单击命令按钮，将显示一个输入对话框，如果在对话框中输入 25，则程序的执行结果是(　　)。

A. 输出 0　　B. 输出 25

C. 输出 45　　D. 无任何输出

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(8) 单击窗体一次，下列程序的输出结果是(　　)。

```
Function func(a,b)
  Static m
  m = m + a + b
  func = m
End Function
Private Sub Command1_Click()
  Dim k,m,p
  k = 4
  m = 1
  p = func(k,m)
  Print p;
  p = func(k, m)
  Print p;
End Sub
```

A. 5　10　　B. 5　5　　C. 6　10　　D. 6　6

4. 编程题

(1) 编制程序实现下列功能，任意输入一个年份，判断该年份是否为闰年。闰年的条件是：年份能被 4 整除但不能被 100 整除，或年份能被 400 整除。要求通过调用函数对某年是否是闰年进行判断，将判断结果直接输出到窗体上即可。

(2) 已知一个升序数组，包含 7 个数组元素。编制两个过程：第一个过程实现数据插入，要求插入一个数据后数组仍然有序；第二个过程实现删除指定的数组元素。

(3) 创建一个应用程序，由 3 个窗体 Form1、Form2 和 Form3 组成。Form1 用于输入用户名和密码(假设用户名和密码分别为“username”和“password”)，输入正确时显示 Form2，连续三次输入错误时显示 Form3。在 Form1 中单击“结束”按钮时结束程序运行；Form2 中用文本框显示“欢迎你使用本系统”，单击“返回”按钮回到 Form1；Form3 中用文本框显示“对不起，请向管理员查询”，单击“退出”按钮结束程序运行。

第 7 章　应用程序界面设计

Visual Basic 6.0 提供的可视化控件和菜单编辑器可以方便地实现应用程序的用户界面设计。使用 Visual Basic 6.0 设计一个实用、友好的应用程序界面，要充分考虑到采用的用户界面类型、需要的窗体、菜单中包含的命令、需要加入的工具栏、对话框的形式以及需要提供的帮助等。

7.1　菜单的设计

菜单是 Windows 用户接口中最重要也是最有特色的元素，使用菜单可以将命令分组，用户可以方便、直观地使用各种命令。

菜单可分为两种类型，即下拉式菜单和弹出式菜单。当启动 Visual Basic 后，单击“文件”菜单所显示的是下拉式菜单，而用鼠标右键单击窗体所显示的菜单就是弹出式菜单。在一个应用系统中，主菜单一般使用下拉式菜单，每一个菜单包含若干个选择项，也叫做子菜单。每一项又可“下拉”出下一级菜单。在 Visual Basic 中，把每个菜单（主菜单或子菜单）看作是一个对象，具备与某些控件相同的属性。每个菜单项相当于一个命令按钮，通常要对其中 Click 事件进行编程，单击菜单项即触发该菜单项的 Click 事件过程。弹出式菜单一般设计为系统中某些功能项的快捷方式，用鼠标右键单击窗体时出现。

7.1.1　下拉式菜单

下拉式菜单是 Windows 应用程序中用得最多的结构。在关闭状态下，它作为菜单栏位于窗口的标题下面，选中某一基本菜单项时，下拉出其相应的子菜单。

1.菜单编辑器

菜单的设计是通过“菜单编辑器”来实现的，用户所看到的每个菜单项都和“菜单编辑器”中定义的一个菜单控件相对应。可以通过 3 种方法进入菜单编辑器。

(1) 选择集成开发环境中主菜单“工具”下菜单编辑器子菜单。

(2) 单击工具栏上的“菜单编辑器”按扭，该按钮的符号为  。

(3) 使用快捷键 Ctrl + E。

菜单编辑器界面如图 7 - 1 所示，界面主要分为两部分：上面部分为菜单控件的属性，该部分可以用来设置菜单项的各种属性，见表 7 - 1；下面部分为菜单项显示区和相关的操作按钮，可以通过该部分的各种操作来设计菜单项，见表 7 - 2。菜单控件列表框列出当前窗体的所有菜单控件。当在标题文本框中键入一个菜单项标题时，该菜单项也会出现在菜单控件列表框中，从列表框中选取一个已存在的菜单控件就可以编辑该控件的属性。

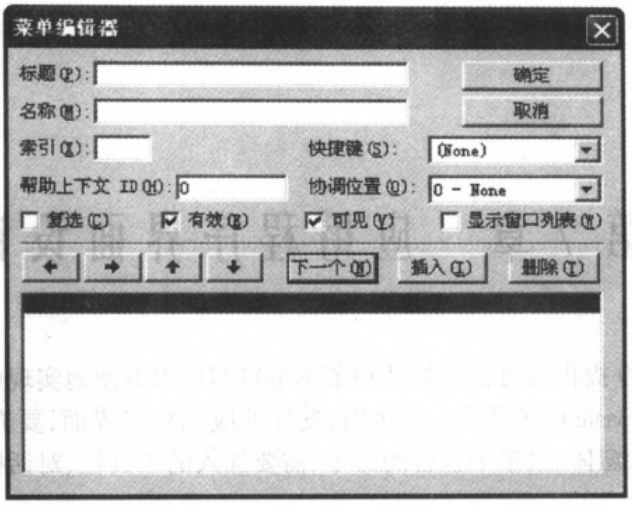


图 7-1 菜单编辑器

表 7-1 菜单项的属性

编号	属性	功能简述
1	标题	所设计的菜单项的标题，是程序运行时显现在菜单上的说明文字，相当于普通控件的 Caption 属性。此外，为了能够通过键盘访问菜单项，可以通过在一个字母前插入 & 符号来设置访问键
2	名称	所设计的菜单项的控件名，也是控件的标识符，这个名称将在程序设计中用来在代码中访问菜单项
3	索引	指定一个数字值来确定菜单项在控件数组中的位置，这个属性是整数
4	快捷键	通过键盘按键的组合来快速访问菜单项
5	帮助上下文 ID	允许为 Context ID 指定惟一数值
6	协调位置	决定是否在容器窗体中显示菜单，以及如何显示
7	复选	决定是否在菜单项的左边显示复选标记
8	有效	决定是否让菜单项对事件做出响应
9	可见	决定是否将菜单项显示在菜单上
10	显示窗口列表	在 MDI 应用程序中，决定一个菜单项是否包含一个打开的 MDI 子窗体列表

表 7-2 设计菜单项的操作

编号	操作	功能简述
1	右箭头	每次单击都把选定的菜单项向右移一个等级，一共可以创建 4 个子菜单等级
2	左箭头	每次单击都把选定的菜单项向左移一个等级，一共可以创建 4 个子菜单等级
3	上箭头	每次单击都把选定的菜单项在同级菜单内向上移动一个位置
4	下箭头	每次单击都把选定的菜单项在同级菜单内向下移动一个位置
5	菜单项显示区	该列表框显示菜单项的分级列表，将子菜单项缩进以指出它们的分级位置或等级
6	下一个	将光标移动到下一个菜单项
7	插入	在列表框的当前选定的菜单项的上方插入一行
8	删除	删除当前选定的菜单项
9	确定	关闭菜单编辑器，并确定菜单项所做的修改
10	取消	关闭菜单编辑器，取消所有修改

2.创建菜单

利用菜单编辑器创建菜单的过程。

【例7-1】 在窗体上创建图7-2所示的应用程序菜单。

图7-2　例7-1程序运行界面

创建菜单控件的步骤：

(1) 打开“菜单编辑器”对话框

(2) 在“标题”文本框中键入“文件”，则标题也同时显示在菜单项显示区中。

(3) 使用Tab键(或鼠标)将焦点移到“名称”栏，并输入“MnuFile”作为菜单控件的名称。

(4) 单击编辑区的“下一个”按钮，菜单项显示区中的光标下移，同时属性区的“标题”栏及“名称”栏被清为空白，光标回到“标题”栏。

(5) 在“标题”栏中键入“新建”，该信息也同时在菜单项显示区显示。

(6) 用Tab键(或鼠标)将焦点移到“名称”栏，并输入“MnuFileNew”作为菜单控件的名称。

(7) 单击编辑区的右箭头(→)，菜单项显示区中的“新建”右移，同时其左侧出现一个内缩符号(…)，表明“新建”是“文件”的下一级菜单。

(8) 单击“快捷键”右端的箭头，显示出各种复合键供选择，从中选出“Ctrl + N”作为“新建”菜单项的热键。此时，在该菜单项的右侧出现“Ctrl + N”。

(9) 再次单击编辑区的“下一个”按钮，菜单项显示区中的光标下移，左端将自动出现内缩符号(…)。同理，可添加“保存”和“退出”等其他菜单项。

(10) 设计完成后的窗口如图7-3所示。此时单击右上角的“确定”按钮，则关闭菜单编辑器窗口。

3.菜单的事件响应程序

菜单设计好后，还要为每个菜单项编写事件代码。其方法如下：

当应用程序的菜单设计完成后，首先关闭菜单编辑器，此时窗体上将显示出所创建的菜单项。这时单击某个菜单项，即可切换到该菜单项的Click事件过程代码窗口中，与编写其它控件事件过程的代码一样，可以在此编写菜单项的Click事件过程的程序代码。例如，单击例7-1中“退出”菜单项，退出程序执行。

```
Private Sub MnuFileExit_Click()
    End
End Sub
```

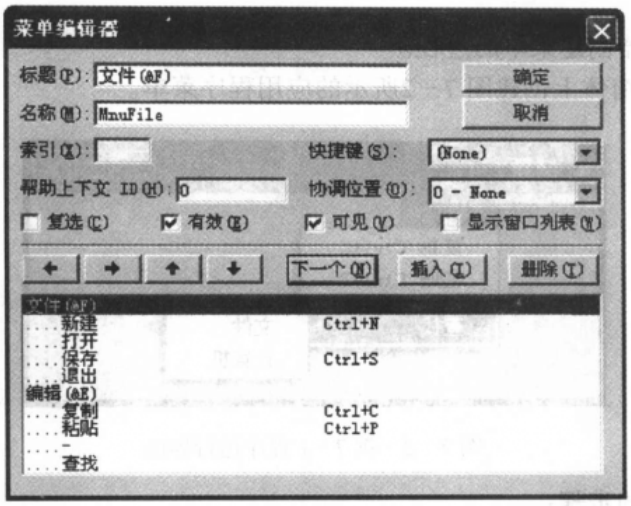


图 7-3 用菜单编辑器设计菜单的过程

7.1.2 弹出式菜单

1.弹出式菜单的设计

弹出式菜单和普通菜单在本质上没有任何区别,其设计与普通菜单完全相同。在运行时,只要菜单至少含有一个子菜单项,就可以作为弹出式菜单。

要设计一个弹出式菜单,必须要保证该菜单项至少具有一个子菜单项。弹出式菜单的"可见"属性即可以是 True 也可以是 False。若为 True,则该菜单项作为菜单栏的一部分显示在菜单栏中,同时也作为弹出式菜单使用;若为 False,则该菜单项不显示在菜单栏中,只作为弹出式菜单使用。

但是在系统中,常常使用弹出式菜单来访问在菜单栏中不常用的选项,因此往往要创建一个不显示在菜单栏里的菜单。这时只需要在设计时使顶级菜单项为不可见(保证在菜单编辑器里的"可见"复选框没有被选上)。当应用程序显示一个弹出式菜单时,指定的顶级菜单的"可见"属性会被忽略。

2.弹出式菜单的调用

为了显示弹出式菜单,需要使用 PopupMenu 方法,可以在大部分响应事件中激活弹出式菜单。但通常都会在鼠标事件(MouseUp)中调用 PopupMenu 方法。

(1) PopupMenu 方法的语法格式为:

[Object.]PopupMenu Menuname[,Flags[,X[,Y[,BoldCommand]]]]

(2) PopupMenu 方法的语法解析见表 7-3。

表 7-3 PopupMenu 方法的语法解析

部 分	描述
Object	可选的。是一个对象表达式,其值为 Form 或者 MDIForm。如果省略 Object,则带有焦点的 Form 对象默认为 Object
Menuname	必须的。指出要显示的弹出式菜单名,指定的菜单项必须至少含一个子菜单项
Flags	可选的。为一个数值或常数,用以指定弹出式菜单的位置和行为。详细设置见表 7-4
X	可选取的。指定显示弹出式菜单的 x 坐标,如果该参数省略,则使用鼠标的横坐标

续表 7-3

部　分	描述
Y	可选取的。指定显示弹出式菜单的 y 坐标，如果该参数省略，则使用鼠标的纵坐标
BoldCommand	可选的。指定弹出式菜单中的菜单控件的名称，用以显示其黑体正文标题，如果该参数省略，则弹出式菜单中没有以黑体出现的控件

(3) Flags 参数指定菜单的位置常数和行为常数见表 7-4。

表 7-4　Flags 参数指定菜单的位置常数和行为常数

	常数位置	值	描述
位置	vbPopupMenuLeftAlign	0	默认值。弹出式菜单的左边定位于 x
	vbPopupMenuCenterAlign	4	弹出式菜单居中于 x
	vbPopupMenuRightAlign	8	弹出式菜单的右边定位于 x
行为	vbPopupMenuLeftButton	0	默认值。仅当使用鼠标左键时，弹出式菜单才响应鼠标单击
	vbPopupMenuRightButton	2	不论使用鼠标左键还是右键，弹出式菜单中的菜单项都响应鼠标单击

7.2　对话框的设计

对话框是应用程序与用户交互的主要途径。Visual Basic 除了使用 InputBox 函数和 MsgBox 函数实现基本的输入/输出功能外，还采用通用对话框控件来调用 Windows 的常用对话框，如“打开”对话框、“另存为”对话框、“颜色”对话框、“字体”对话框、“打印”对话框和“帮助”对话框等。

7.2.1　CommonDialog 控件

CommonDialog 控件是常用的一个控件，它可以提供打开、另存为、颜色、字体、打印、帮助等几种类型的标准对话框。该控件属于 ActiveX 控件而不是 Visual Basic 的标准控件，用户使用之前，需要先将其引入控件箱。具体操作如下：

(1) 选择“部件”菜单项，或按快捷键 Ctrl + T，弹出如图 7-4 所示的“部件”对话框。

(2) 在“部件”对话框中选中“Microsoft Common Dialog Control 6.0”。

CommonDialog 控件的属性是与不同的对话框紧密相关的，有些属性只适合于某一类对话框，有些属性在不同的对话框中属性是有差别的，下面分类列出与不同对话框相关联的属性用法。

(1) 与 ShowOpen、ShowSave 方法相关的属性见表 7-5。

表 7-5　CommonDialog 控件中与 ShowOpen 、ShowSave 方法相关的属性

编号	属性	功能简述
1	FileName	返回或设置所选文件的路径和文件名，如果在使用 Show 方法前使用 FileName 属性，则设定对话框的默认文件名；如果在使用 Show 方法后使用 FileName 属性，则返回选择的文件名
2	DefaultExt	为该对话框返回或设置缺省的文件扩展名

(2) 与 ShowColor 方法相关的属性如下：

Color 属性为选定的颜色。要使用此属性，必须先将 Flags 属性设置为 cdlCFEffects。

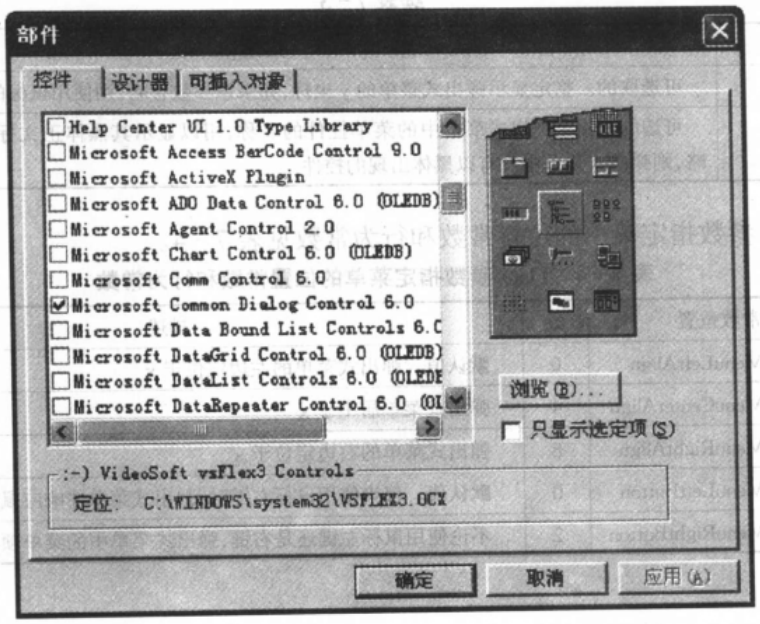


图 7－4　添加 ActiveX 控件的“部件”对话框

(3) 与 ShowFont 方法相关的属性见表 7－6。

表 7－6　CommonDialog 控件中与 ShowFont 方法相关的属性

编号	属性	功能简述
1	Color	选定的颜色。为使用此属性,必须先将 Flags 属性设置为 cdlCFEffects
2	FontBold	是否选定粗体
3	FontItalic	是否选定斜体
4	FontStrikethru	是否选定删除线
5	FontUnderline	是否选定下划线
6	FontName	选定的字体名
7	FontSize	选定的字体大小

(4) 与 ShowHelp 方法相关的属性见表 7－7。

表 7－7　CommonDialog 控件中与 ShowHelp 方法相关的属性

编号	属性	功能简述
1	HelpCommand	返回或设置需要的联机帮助的类型
2	HelpFile	确定帮助文件的路径和文件名

1.“打开”对话框

“打开”对话框用于指定用户要使用的文件名和位置。“打开”对话框如图 7－5 所示。

显示文件对话框可以有两种方法:

(1) 设置对话框控件的属性。可以通过设置对话框的 Action 属性来显示不同的对话框,如果要显示打开对话框,则将 CommonDialog 控件对象的 Action 属性设置为 1。例如:

CommonDialog1.Action=1

(2) 使用对话框控件的方法。对话框控件提供了显示 Windows 标准对话框的方法,如果要显示打开对话框,则可以使用 ShowOpen 方法。例如:

CommonDialog1.ShowOpen

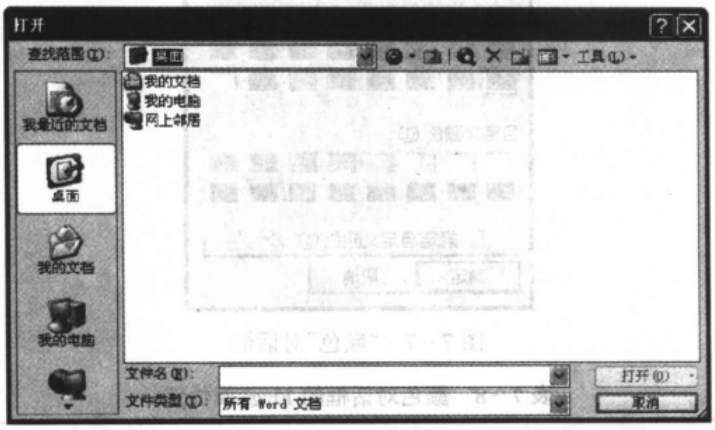


图 7-5　“打开”对话框

2.“另存为”对话框

“另存为”对话框用于指定要保存的文件名和位置。“另存为”对话框如图 7-6 所示。

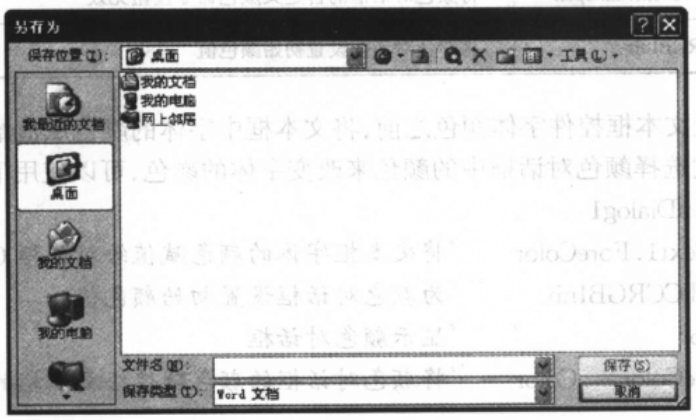


图 7-6　“另存为”对话框

同理,显示“另存为”对话框的方法也有两种:将 Action 属性设置为 2 或使用 ShowSave 方法。

3.“颜色”对话框

“颜色”对话框用来设置窗体或控件中有关颜色的属性,如前景色、背景色,“颜色”对话框如图 7-7 所示。用户可以使用其中的基本颜色,也可以自己调色。当用户选中某一种颜色后,该颜色值赋给 Color 属性。

将对话框控件设置为“颜色”对话框的方法与“打开”对话框基本相同。显示“颜色”对话框的方法也有两种:将 Action 属性设置为 3 或使用 ShowColor 方法。

“颜色”对话框的 Flags 属性设置值见表 7-8 所示。

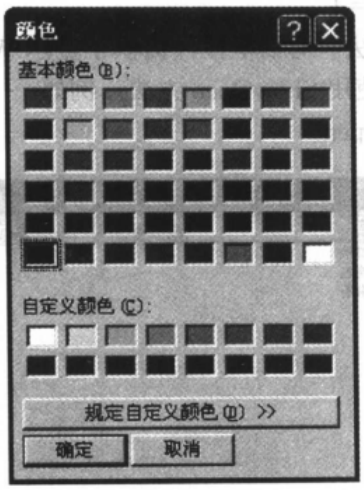


图 7-7 “颜色”对话框

表 7-8 颜色对话框的 Flags 属性值

设置值	对应的 Visual Basic 常数	描 述
&H2	cdlCCFullOpen	显示全部的颜色对话框,包括定义自定义部分
&H8	cdlCCShowHelpButton	指定颜色对话框显示帮助按钮
&H4	cdlCCPreventFullOpen	将颜色对话框的自定义颜色命令按钮无效
&H1	cdlCCRGBInit	为颜色对话框设置初始颜色值

例如,在设置文本框控件字体颜色之前,将文本框中字体的颜色赋值给通用对话框 Color 属性,然后通过选择颜色对话框中的颜色来改变字体的颜色,可以使用下面的代码。

```
With CommonDialog1
  .Color = Text1.ForeColor      '将文本框字体的颜色赋值给对话框 Color 属性
  .Flags = cdlCCRGBInit         '为颜色对话框设置初始颜色值
  .ShowColor                    '显示颜色对话框
  Text1.ForeColor = .Color      '将颜色对话框的颜色赋值给文本框的字体颜色
End With
```

4.“字体”对话框

“字体”对话框用于设置字体的样式。用户可以像使用 Windows 标准字体对话框一样使用自己应用程序中的字体对话框设置字体的样式。“字体”对话框如图 7-8 所示。

显示字体对话框的方法是 ShowFont,也可以通过将对话框控件的 Action 属性设置为 4 来显示字体对话框。

同“文件”对话框一样,要设置“字体”对话框的样式,需要使用 Flags 属性。在显示字体对话框之前,必须先将 Flags 属性设置为 cdlCFScreenFonts(使用屏幕字体)、cdlCFPrinterFonts(使用打印机字体)或 cdlCFBoth(两者都可以)。否则,会产生字体不存在的错误。

通过字体对话框的返回值可以设置字体的样式,字体对话框的返回值见表 7-9。

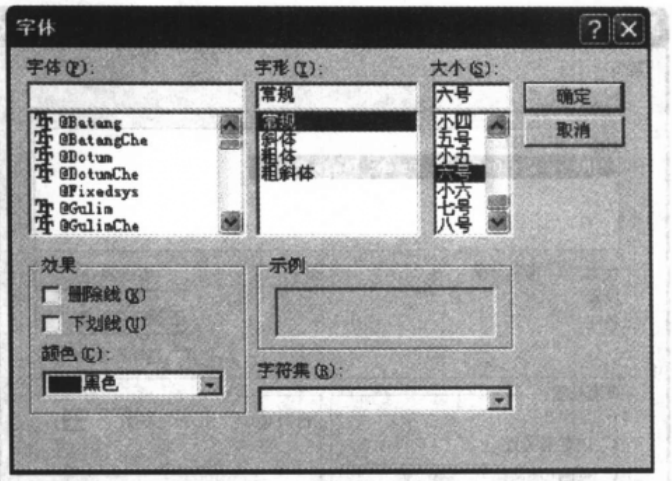


图7-8　“字体”对话框

表7-9　字体对话框的属性值

字体对话框属性	描　述
FontName	返回字体名称
FontSize	返回以点为单位的字体的大小
FontBold	返回字体是否为粗体
FontItalic	返回字体是否为斜体
FontUnderLine	返回字体是否带下划线
FontStrikethru	返回字体是否带删除线

例如，要设置文本框中字体的样式，可以通过显示“字体”对话框后，将“字体”对话框的返回值传递给文本框控件的字体属性来实现，下面的代码完成了Text1字体属性的设置。

```
With CommonDialog1
    Text1.Font.Name = .FontName                  '设置字体名称
    Text1.Font.Bold = .FontBold                  '设置粗体属性
    Text1.Font.Italic = .FontItalic              '设置斜体属性
    Text1.Font.Size = .FontSize                  '设置字体大小
    Text1.Font.Strikethrough = .FontStrikethru   '设置字体是否有删除线
    Text1.Font.Underline = .FontUnderline        '设置字体是否有下划线
    Text1.ForeColor = .Color                     '设置字体颜色
End With
```

5.“打印”对话框

“打印”对话框用来设置打印机的属性以及为打印机处理指定相应的选项，如打印范围、数量等。“打印”对话框如图7-9所示。

显示“打印”对话框方法是ShowPrinter，也可以通过将对话框控件的Action属性设置为5来显示打印对话框。

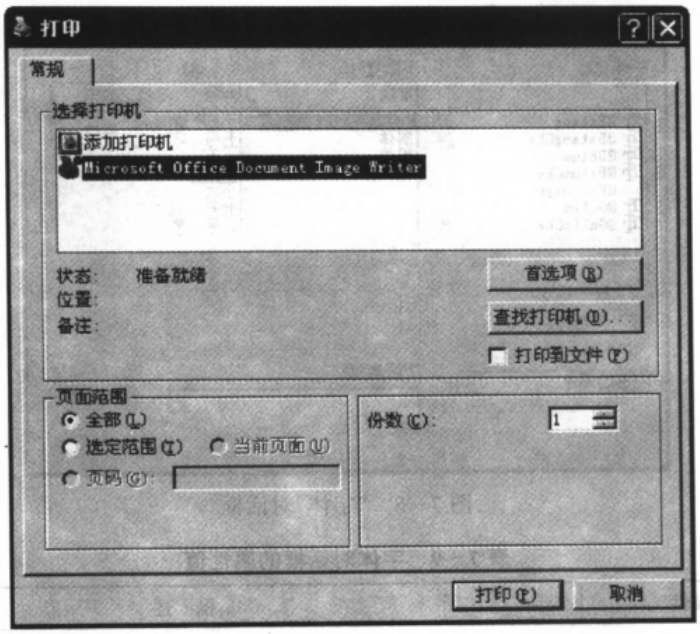


图 7-9 “打印”对话框

6.“帮助”对话框

“帮助”对话框实际上是通过运行 WinHlp.exe 来显示指定的帮助文件,使用的方法是 ShowHelp。在使用 ShowHelp 方法前,必须将对话框控件的 HelpFile 属性设置为正确格式化的 Windows 帮助文件(*.Hlp)的名称和位置,并将 HelpCommand 属性设置为一个常数以告诉 Help 引擎要提供何种类型的帮助。否则,WinHlp.exe 就无法显示对应的帮助文件。

例如,下面的代码用来显示 Visual Basdic 6.0 的帮助文件。

```
With CommonDialog1
  .HelpFile="VB6.hlp"                    '设置帮助文件
  .HelpCommand=cdlHelpContents           '设置要显示帮助目录的主题
  .ShowHelp                              '显示 Visual Basic 帮助目录的主题
End With
```

7.2.2 通用对话框的应用

【例 7-2】 利用通用对话框设计一个简单的文本编辑器,程序运行界面如图 7-10 所示。

程序设计步骤如下:

(1) 新建一个“标准 EXE”工程。

(2) 建立程序用户界面。在窗口中添加一组名为 Command1 的命令按钮控件数组,将数据元素的 Caption 属性分别设置为“新建”、“打开”、“另存”、“字体”、“颜色”和“退出”;再添加一个通用对话框控件和一个 RichTextBox 控件。

(3) 进入代码编辑窗口,编写如下事件过程。

```
Private Sub Command1_Click(Index As Integer)
  Dim t As Integer
```

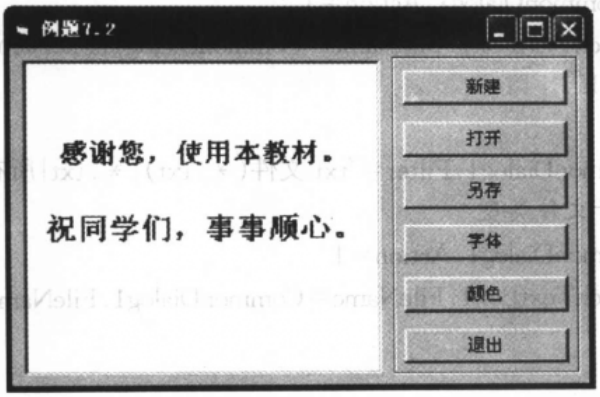


图7-10　例7-2程序运行界面

```
Select Case Index
  Case 0
    '如果有操作,则提示用户是否放弃当前内容
    If Trim(RichTextBox1.Text)<>"" Then
      t=MsgBox("你是否要保存当前编辑",vbYesNo+vbInformation+vbDe_
      faultButton2)
      If t=vbYes Then
        '如果选 Yes,则先保存当前编辑的内容
        CommonDialog1.Filter="txt 文件(*.txt)|*.txt"
        CommonDialog1.Action=2
        RichTextBox1.SaveFile(CommonDialog1.FileName)
        '再清空编辑区
        RichTextBox1.Text=""
      Else
        '否则直接清空编辑区
        RichTextBox1.Text=""
      End If
    End If
  Case 1
    '如果有操作,则提示用户是否放弃当前内容
    If Trim(RichTextBox1.Text)<>"" Then
      t=MsgBox("你是否要放弃当前编辑",vbYesNo+vbInformation+vbDe _
      faultButton2)
      '如果选 Yes 则放弃当前编辑
      If t=vbYes Then
        CommonDialog1.Filter="txt 文件(*.txt)|*.txt|所有文件|*.*"
        '打开文件类型
```

```
            CommonDialog1.Action = 1
            RichTextBox1.FileName = CommonDialog1.FileName
          End If
        Else
          CommonDialog1.Filter = "txt 文件(*.txt)|*.txt|所有文件|*.*"
          '打开文件类型
          CommonDialog1.Action = 1
            RichTextBox1.FileName = CommonDialog1.FileName
        End If
      Case 2
        CommonDialog1.Filter = "txt 文件(*.txt)|*.txt"
        CommonDialog1.Action = 2
        RichTextBox1.SaveFile (CommonDialog1.FileName)
      Case 3
        CommonDialog1.Flags = cdlCFBoth Or cdlCFEffects
        CommonDialog1.ShowFont
        RichTextBox1.SelFontName = CommonDialog1.FontName
        RichTextBox1.SelFontSize = CommonDialog1.FontSize
        RichTextBox1.SelBold = CommonDialog1.FontBold
        RichTextBox1.SelItalic = CommonDialog1.FontItalic
        RichTextBox1.SelUnderline = CommonDialog1.FontUnderline
        RichTextBox1.SelStrikeThru = CommonDialog1.FontStrikethru
        RichTextBox1.SelColor = CommonDialog1Color
      Case 4
        CommonDialog1.Action = 3
        RichTextBox1.SelColor = CommonDialog1.Color
      Case 5
        End
  End Select
End Sub
```

7.3 工具栏的设计

工具栏是 Windows 应用程序界面常见的组成部分。工具栏包含一组图像按钮，是用户访问应用程序的最常用功能和命令的图像集合。

在 Visual Basic 程序设计中，为窗体添加工具栏应使用 ToolBar 控件和 ImageList 控件。Visual Basic 专业版和企业版中都有 ToolBar 控件和 ImageList 控件。ToolBar 控件和 ImageList 控件是 ActiveX 控件，在使用时必须先添加到工具箱中。

创建工具栏的步骤如下：

(1) 添加 ToolBar 控件和 ImageList 控件。

选择“部件”菜单项,或按快捷键 Ctrl + T,打开“部件”对话框,在“部件”对话框中选中“Microsoft Windows Common Control 6.0”,单击“确定”按钮,则 ToolBar 控件和 ImageList 控件出现在工具箱中。

(2) 创建 ImageList 控件作为使用图形的集合。创建 ToolBar 控件,且将 ToolBar 控件与 ImageList 控件相关联,创建 Button 对象。

(3) 在 ButtonClick 事件中添加代码。

7.3.1　ImageList 控件

ImageList 控件的作用就像图像的储藏室。ImageList 控件不能独立使用,它需要 ToolBar 控件(或 ListView、TabStrip、Header、ImageCombo 和 TreeView 控件)显示所存储的图像。

ImageList 控件的 ListImage 属性是对象的集合。每个对象可以存放图像文件,图像文件类型有.bmp、.cur、.ico、.jpg 和.gif,且可通过索引(Index)或关键字(Key)引用每个对象。

在设计时,可在 ImageList 控件属性页中添加图像,按照顺序将需要的图像插入 ImageList 中。例如,在窗体上创建 ImageList1 后,用鼠标右键单击 ImageList1 控件对象,出现弹出式菜单,选择“属性”命令,则出现属性页,选择“图像”选项卡,在“图像”选项卡中插入图片,如图 7-11 所示。其中:

“索引”表示每个图像的编号,在 ToolBar 的按钮中引用。

“关键字”表示每个图像的标识名,在 ToolBar 的按钮中引用。

“图像数”表示已插入的图像数目。

“插入图片”用于插入新图像。

“删除图片”用于删除选中的图像。

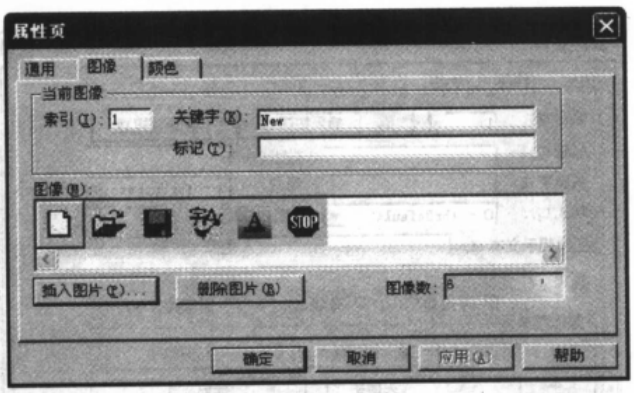


图 7-11　ImageList 控件属性页

7.3.2　ToolBar 控件

1. 工具栏连接图像

(1) 双击工具箱中的 ToolBar 按钮,向窗体中添加 ToolBar 控件,它是一条灰色的区域。

(2) 右击 ToolBar 控件,打开 ToolBar 控件的属性对话框,选择“通用”选项卡,如图 7-12 所示。其中:

“图像列表”表示与 ImageList 控件的连接。

“可换行的”表示当工具栏的长度不能容纳所有的按钮时，在下一行显示，否则剩余部分不显示。

“样式”是 Visual Basic 6.0 新增的功能，表示工具栏的风格。

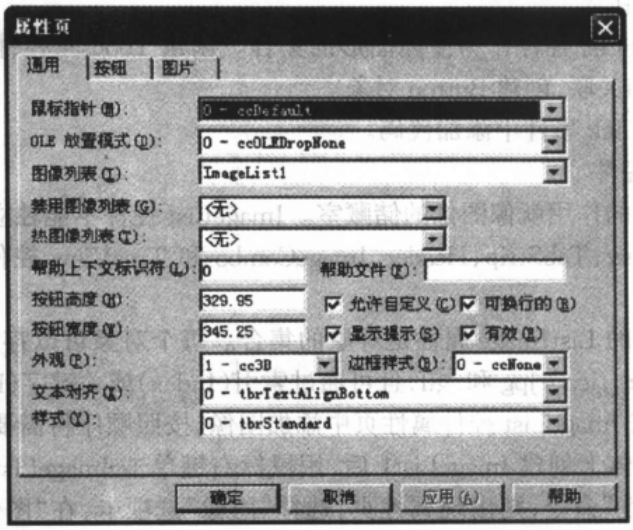


图 7-12 ToolBar 控件“通用”选项卡

2. 工具栏增加按钮

(1) 将“属性页”对话框的操作切换到“按钮”选项卡，用来创建工具栏中的按钮，如图 7-13 所示。其中：“样式”用于选择按钮样式，共 6 种，含义见表 7-10；“图像”表示 ImageList 对象中的图像，它的值可以是图中的关键字或索引值；“值”表示按钮的状态，有按下(tbrPressed)和没按下(tbrUnpressed)，对样式 1 和样式 2 有用。

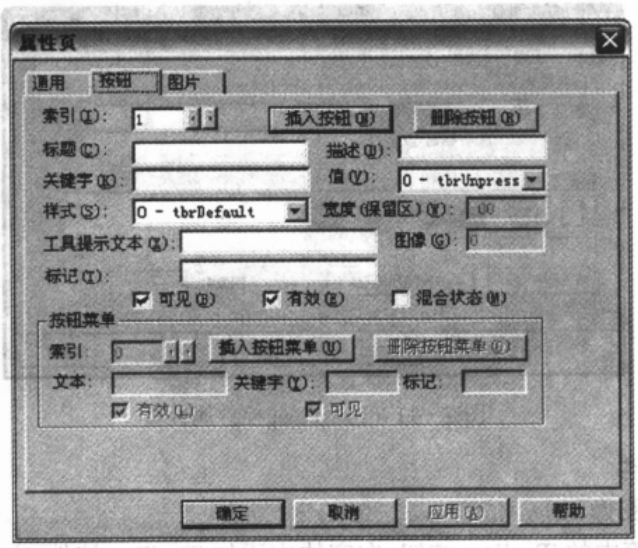


图 7-13 ToolBar 控件“按钮”选项卡

(2) 单击“插入按钮”，插入第 1 个按钮并设置它的属性值。索引和关键字两个属性都可以惟一标识一个按钮，在按钮的属性设置中，两者至少一个不为空。若“图像”为 1，表示

该按钮使用 ImageList 中的第 1 个图像。

表 7-10　按钮样式

值	常　数	说　明
0	tbrDefault	普通按钮。按钮按下后恢复原态
1	tbrCheck	开关按钮。按钮按下后将保持按下状态
2	tbrButtonGroup	编组按钮。一组按钮同时只能一个有效
3	tbrSeparator	分隔按钮。把左右的按钮分隔
4	tbrPlaceholder	占位按钮。以便安放其他按钮，可设置按钮宽度
5	tbrDropdown	菜单按钮。具有下拉式菜单

(3) 单击“插入按钮”，插入第 2 个按钮并设置它的属性值，以此类推。

3. ToolBar 控件

ToolBar 控件常用的事件有两个：ButtonClick 和 ButtonMenuClick。前者对应按钮样式为 0～2，后者对应样式为 5 的菜单按钮。

工具栏上的按钮是控件数组，单击工具栏上的任何一个按钮都会发生 ButtonClick 或 ButtonMenuClick 事件。可以利用数组的索引或关键字来区分所单击的按钮，然后再使用 Select Case 语句编写按钮的功能代码。现以 ButtonClick 事件为例如下：

(1) 用索引 Index 确定按钮，其形式为：

```
Private Sub Toolbar1 _ ButtonClick(ByVal Button As MSComctlLib.Button)
  Select Case Button.Index
    Case 1
      …                    '按第 1 个按钮，执行相应的过程
    Case 2
      …                    '按第 2 个按钮，执行相应的过程
      …                    '其他情况处理
  End Select
End Sub
```

(2) 用关键字 Key 来确定按钮(若第一个按钮的关键字是 New，第二个按钮的关键字是 Open)。

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
  Select Case Button.Key
    Case "New"
      …                    '按第 1 个按钮，执行相应的过程
    Case "Open"
      …                    '按第 2 个按钮，执行相应的过程
      …                    '其他情况处理
  End Select
End Sub
```

比较上面两种程序的形式，如果使用 Index(索引)，当按钮有增删时，Index 的值也随之

改变,要保证程序执行的正确就要修改程序。但是如果使用 Key(关键字),程序的可读性好,而且当按钮有增删时,使用关键字不会影响程序。

【例 7-3】 设计一个带工具栏的简单的文本编辑器,界面如图 7-14 所示。

程序设计步骤如下:

(1)新建一个"标准 EXE"工程。

(2) 通过"部件"对话框向工具箱添加通用对话框控件、RichTextBox 控件、ImageList 控件和 ToolBar 控件。

(3) 建立程序用户界面。在窗口中添加通用对话框控件、RichTextBox 控件、ImageList 控件和 ToolBar 控件。

(4) 在 ImageList 控件中添加图像。

(5) 为 ToolBar 控件连接图像。

(6) 为 ToolBar 控件增加按钮。

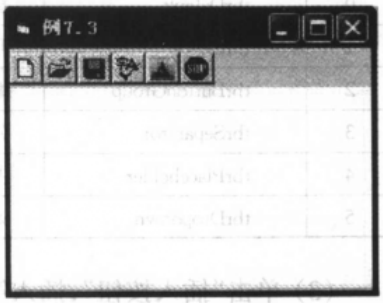


图 7-14 例 7-3 程序运行界面

(7) 进入代码编辑窗口,编写如下事件过程。

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
  Select Case Button.Index
    Case 1
      '如果有操作,则提示用户是否放弃当前内容
      If Trim(RichTextBox1.Text)<>"" Then
        t=MsgBox("你是否要保存当前编辑",vbYesNo+vbInformation+_
        vbDefault Button2)
        If t=vbYes Then                    '如果选 Yes 则先保存当前编辑的内容
          CommonDialog1.Filter="txt 文件(*.txt)|*.txt"
          CommonDialog1.Action=2
          RichTextBox1.SaveFile(CommonDialog1.FileName)
          RichTextBox1.Text=""   '再清空编辑区
        Else
          RichTextBox1.Text=""   '否则直接清空编辑区
        End If
      End If
    Case 2
      '如果有操作,则提示用户是否放弃当前内容
      If Trim(RichTextBox1.Text)<>"" Then
        t=MsgBox("你是否要放弃当前编辑", vbYesNo+vbInformation+_
        vbDefaultButton2)
        If t=vbYes Then                    '如果选 Yes 则放弃当前编辑
          CommonDialog1.Filter="txt 文件(*.txt)|*.txt|所有文件|*.*"
          '打开文件类型
          CommonDialog1.Action=1
          RichTextBox1.FileName=CommonDialog1.FileName
```

```
            End If
        Else
            CommonDialog1.Filter="txt 文件(*.txt)|*.txt|所有文件|*.*"
            '打开文件类型
            CommonDialog1.Action=1
            RichTextBox1.FileName=CommonDialog1.FileName
        End If
    Case 3
        CommonDialog1.Filter="txt 文件(*.txt)|*.txt"
        CommonDialog1.Action=2
        RichTextBox1.SaveFile (CommonDialog1.FileName)
    Case 4
        CommonDialog1.Flags=cdlCFBoth Or cdlCFEffects
        CommonDialog1.ShowFont
        RichTextBox1.SelFontName=CommonDialog1.FontName
        RichTextBox1.SelFontSize=CommonDialog1.FontSize
        RichTextBox1.SelBold=CommonDialog1.FontBold
        RichTextBox1.SelItalic=CommonDialog1.FontItalic
        RichTextBox1.SelUnderline=CommonDialog1.FontUnderline
        RichTextBox1.SelStrikeThru=CommonDialog1.FontStrikethru
        RichTextBox1.SelColor=CommonDialog1Color
    Case 5
        CommonDialog1.Action=3
        RichTextBox1.SelColor=CommonDialog1.Color
    Case 6
        End
  End Select
End Sub
```

7.4　多重窗体的设计

用户界面样式主要有单文档界面和多文档界面两种。单文档界面的一个示例就是 Microsoft Windows 中的 NotePad(记事本)应用程序。在 NotePad 中,只能打开一个文档,要打开另一个文档时,必须先关上已打开的文档。但绝大多数基于 Windows 的大型应用程序都是多文档界面,如 Microsoft Excel 和 Microsoft Word 等。多文档界面允许同时打开多个文档,每一个文档都显示在自己的被称为子窗口的窗口中。多文档界面由父窗口和子窗口组成,一个父窗口可包含多个子窗口,子窗口最小化后将以图标形式出现在父窗口中,而不会出现 Windows 的任务栏中。当最小化父窗口时,所有的子窗口也被最小化,只有父窗口的图标出现在任务栏中。在 Visual Basic 中,父窗口就是 MDI 窗体,子窗口是指 MDChild 属性

为 True 的普通窗体。

7.4.1 创建 MDI 窗体

用户要建立一个 MDI 窗体，可以选择“工程”菜单中的“添加 MDI 窗体”命令，会弹出“添加 MDI 窗体”对话框，选择“新建 MDI 窗体”或“现存 MDI 窗体”，再选择“打开”按钮。

一个应用程序只能有一个 MDI 窗体，但可以有多个 MDI 子窗体。如果工程已经有了一个 MDI 窗体，则不可用“工程”菜单上的“添加 MDI 窗体”命令。

7.4.2 加入 MDI 子窗体

创建子窗体只要把一个普通窗体的 MDIChild 属性设置为 True 即可。有两种方法可以添入 MDI 子窗体。

1. 将一般窗体变为 MDI 子窗体

一般在启动 Visual Basic 后，系统自动创建了一个窗体 Form1，在建立了 MDI 窗体之后，Form1 窗体还不是 MDI 中的成员，要让它成为子窗体，必须将其 MDIChild 属性设置为 True。

2. 添加 MDI 子窗体

选择“工程”菜单中的“添加窗体”命令，像前面添加 MDI 窗体一样，出现添加窗体时，单击“窗体”，在屏幕上会出现一个新的窗体，将其 MDIChild 属性设置为 True。如果要建立多个子窗体，只要重复进行上述的操作即可。

7.4.3 MDI 窗体与子窗体的交互操作

1. 加载 MDI 窗体及子窗体

程序运行后，系统会自动加载并显示 MDI 窗体，但其子窗体不会自动加载。因此，需要在父窗体的 Load 事件中加入显示子窗体的代码。例如，以下代码加载并显示两个 MDI 子窗体 Form1 和 Form2：

```
Private Sub MDIForm1_Load()
  Form1.Show
  Form2.Show
End Sub
```

MDI 窗体有 AutoShowChildren 属性，决定是否自动显示子窗体。如果它被设置为 True，则当改变子窗体属性(如 Caption)后，会自动显示该子窗体，不再需要 Show 方法；如果 MDI 窗体有 AutoShowChildren 属性设置为 False，则改变子窗体属性后，不会自动显示该子窗体，子窗体处于隐藏状态直至用 Show 方法把它们显示出来。MDI 子窗体没有 AutoShowChildren 属性。

2. 关闭 MDI 窗体

和普通窗体一样，关闭 MDI 窗体的代码如下：

```
UnloadMDI 窗体名
或
Unload me
```

系统在执行该代码后，将先触发 QueryUnload 事件，如果需要保存有关信息及其他处理，可在该事件代码中完成。然后卸载各子窗体，最后卸载 MDI 窗体。

习题七

1. 简答题

(1) ToolBar 控件的作用是什么？ImageList 控件的作用是什么？如何使两个控件连接？

(2) 当要在 ToolBar 控件中增加一个按钮，如何实现？当要修改 ToolBar 控件某按钮的图像，如何实现？

(3) 在使用通用对话框控件调用“打开”或“另存为”对话框时如何设置其 Filter 属性？

(4) 弹出式菜单与下拉式菜单在使用时有何区别？如何使一个下拉式菜单成为弹出式菜单？

2. 填空题

(1) 单击工具栏上的按钮会发生 ButtonClick 事件或 ButtonMenuClick 事件，可以利用(　　)和(　　)来识别被单击的按钮。

(2) 工具栏的制作方法有两种，它们分别是(　　)和(　　)。

(3) 在菜单编辑器中建立一个菜单，其名称为 mmuEdit，Visible 属性为 False。程序运行后，如果用鼠标右键单击窗体，则弹出与 mmuEdit 相应的菜单。以下是实现上述功能的程序，请填空。

```
Private Sub Form (    ) (Button As Integer, Shift As Integer, X As Single, Y As Single)
  If Button = 2 Then
    (    ) mmuEdit
  End If
End Sub
```

【出处】2004 年 4 月计算机等级考试二级 Visual Basic

(4) 在菜单编辑器中建立一个菜单，名为 pmenu，用下面的语句可以把它作为弹出式菜单弹出，请填空。

```
Form1.(    ) pmenu
```

【出处】2002 年 9 月计算机等级考试二级 Visual Basic

(5) 如果要将某个菜单项设计为分隔线，则该菜单项的标题应设置为(　　)。

【出处】2006 年 4 月计算机等级考试二级 Visual Basic

3. 选择题

(1) 在用菜单设计器设计菜单时，必须输入的是(　　)。

A. 快捷键　　B. 标题　　C. 索引　　D. 名称

(2) 在下列关于菜单的说法中，错误的是(　　)。

A. 每个命令都是一个控件，与其他控件一样有自己的属性和事件

B. 除了 Click 事件外，命令还可以响应其他事件

C. 命令的快捷键不能任意设置

D. 在程序执行过程中，如果命令的 Enabled 属性值为 False，则该命令变灰，不能被用户选择

(3) 下列关于通用对话框的叙述中，错误的是(　　)。

A. CommonDialog1.ShowFont　　'显示字体对话框

B. 在打开对话框中，用户选择的文件名及其路径可以通过 FileTitle 属性返回

C. 在打开对话框中，用户选择的文件名及其路径可以通过 FileName 属性返回

D. 通用对话框可以用来制作和显示帮助对话框

(4) 工具栏上的按钮样式有(　　)种。

A. 3　　B. 4　　C. 5　　D. 6

(5) 多文档界面具有的特性是(　　)。

A. 子窗体可以被拖动到父窗体之外

B. 子窗体被限制在父窗体之中

C. 父窗体有自己的菜单,子窗体不可以有自己的菜单

D. 当最小化窗体时,MDI 的图标显示在任务栏中

(6) 加载子窗体时,(　　)。

A. 其父窗体会自动加载但不显示

B. 其父窗体不会自动加载

C. 其父窗体会自动加载并显示

D. 其父窗体会自动加载,还需要用 Show 方法显示

(7) 如果要在菜单中添加一个分隔线,则应将其 Caption 属性设置为(　　)。

A. =　　B. *　　C. &　　D. —

【出处】2003 年 9 月计算机等级考试二级 Visual Basic

(8) 以下叙述中错误的是(　　)。

A. 下拉式菜单和弹出式菜单都用菜单编辑器建立

B. 在多窗体程序中,每个窗体都可以建立自己的菜单系统

C. 除分隔线外,所有菜单项都能接收 Click 事件

D. 如果把一个菜单项的 Enabled 属性设置为 False,则该菜单项不可见

【出处】2004 年 4 月计算机等级考试二级 Visual Basic

(9) 假定所列的菜单结见表 7－11。

表 7－11　菜单结构

标题	名称	层次
显示	appear	1(主菜单)
大图标	bigicon	2(子菜单)
小图标	smallicon	2(子菜单)

要求程序运行后,如果单击菜单项“大图标”,则在该菜单项前添加一个“?”。以下正确的事件过程是(　　)。

A.
```
Private Sub bigicon_Click()
    bigicon.Checked = False
End Sub
```

B.
```
Private Sub bigicon_Click()
    Me.appear.bigicon.Checked = True
End Sub
```

C.
```
Private Sub bigicon_Click()
    bigicon.Checked = True
End Sub
```

D.
```
Private Sub bigicon_Click()
    appear.bigicon.Checked = True
End Sub
```

【出处】2005 年 9 月计算机等级考试二级 Visual Basic

(10) 假定通用对话框的名称为 CommonDialog1,命令按钮的名称为 Command1,则单击命令按钮后,能

使打开的对话框的标题为“New Title”的事件过程是(　　)。

A. Private Sub Command1_Click()
　　CommonDialog1. DialogTitle = "New Title"
　　CommonDialog1. ShowPrinter
　End Sub

B. Private Sub Command1_Click()
　　CommonDialog1. DialogTitle = "New Title"
　　CommonDialog1. ShowFont
　End Sub

C. Private Sub Command1_Click()
　　CommonDialog1. DialogTitle = "New Title"
　　CommonDialog1. ShowOpen
　End Sub

D. Private Sub Command1_Click()
　　CommonDialog1. DialogTitle = "New Title"
　End Sub

【出处】2005 年 9 月计算机等级考试二级 Visual Basic

4. 编程题

设计一个带有工具栏的文本编辑器,如图 7－15 所示,功能包括:

(1) 有菜单和工具栏,以方便使用。

(2) 支持打开、保存文件的功能,支持常用的编辑操作。

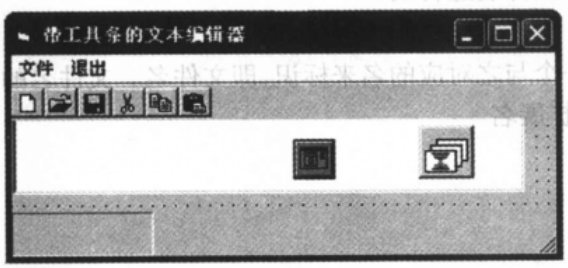


图 7－15　程序界面

第 8 章　文件操作

在应用程序中，输入和输出操作是重要的基本操作。Visual Basic 的输入、输出即可以在标准的输入设备(如键盘)和输出设备(如显示器、打印机)上进行，也可以在其他外部设备(如磁盘、磁带等存储介质等)上进行。而记录在外部存储介质上的数据是以文件的形式存储的，因此对于非标准的输入输出通常称为文件处理。Visual Basic 具有较强的文件处理功能，其提供了多种处理文件的方法。本章将详细介绍文件系统、3 种不同类型文件(顺序文件、随机文件和二进制文件)的处理方法以及 Visual Basic 提供的文件系统控件。

8.1　文件的基础知识

由于计算机内存的特点，程序中所需的数据不可能一直存放在内存中。为了能够“永久地”存储数据信息，就需要将数据以文件的形式存储在相关的外部存储介质(如磁盘)中。所谓“文件”就是指记录在外部介质上的数据集合。文件通过与之对应的文件名表示，而对文件的访问是通过文件的文件名进行的。

1. 文件名

每一个文件都用一个与之对应的名来标识，即文件名。文件名的一般格式为：

文件基本名.扩展名

例如：

mydata.txt

说明：

(1) 文件的扩展名往往用来说明文件的类型。如文本文件为“.txt”、可执行文件为“.exe”。

(2) 文件名中还可以带有驱动器和路径信息。例如：

C:\datainfo\ydata.txt

即是指 C 盘 datainfo 文件夹(目录)中的“mydata.txt”文件。

2. 文件结构

为了有效地对数据进行存储和读取，文件中的数据必须以某种特定的格式去存储，这种特定的格式称为文件的结构。在 Visual Basic 中，对于磁盘文件的处理都是以“记录”的方式进行，即一个文件是由一个或多个记录组成的。而文件中的记录由字段组成，字段又是由字符组成的。

(1) 文件(File)：存储在外部介质上数据的集合，一个文件以一个或多个记录组成。

(2) 记录(Record)：由一组相关的字段组成，通常用来表示一个具体实体。

(3) 字段(Field)：亦称为“域”或数据项，由若干个字符组成，用来表示一项数据。

(4) 字符(Character):是构成文件的最基本单位。

3.文件的读写操作

读写操作是对文件的两种主要操作。在文件的操作中,把内存中的数据传输到相应的外部设备(如磁盘),并作为文件存放的操作叫做写数据;把数据文件中的数据传输到内存中的操作叫做读数据。一般来说,在主存与外设的数据传输中,由主存到外设叫做输出或写,由外设到主存叫做输入或读。

在 Visual Baisc 中外部文件同内存之间的信息交换是通过文件缓冲区进行的。当打开一个文件时,系统自动为正在使用的文件开辟内存缓冲区。当从磁盘文件中读取数据至内存时,并不直接从文件中读取数据,而是从输入文件缓冲区中读取。只有当输入缓冲区中无数据时,则再从文件中读取一批数据存入缓冲区。同样,当向文件写入数据时,先将数据输出至输出缓冲区,当输出缓冲区满(或文件正常关闭等)时,再一次性地将输出缓冲区中的数据写入文件中。

4.文件的种类

按照不同分类的标准,文件可分为几种不同的类型。对不同组织结构的文件,其存取方法也不同。

(1) 按数据性质分类可以将文件分为程序文件(Program File)和数据文件(Data File)两大类。

程序文件:存放计算机执行程序的文件,包括源文件、可执行文件等。

数据文件:用来存放利用程序进行处理的数据,这类数据必须通过程序进行存取和管理。

(2) 按数据的存取方式和结构分类可以将文件可分为顺序文件和随机文件两类。

顺序文件(Sequential Access File):一般为普通文本文件。这种文件的组织结构比较简单,文件中的记录按顺序存放。记录的长度也可按需要变化,一行一条记录,记录可长可短,以"换行"字符为分隔符号。在这种文件中,只知道第一个记录的存放位置,其他记录的位置则无从知道。当要在文件中查找某个数据时,只能从文件头开始,一个记录一个记录地顺序读取,直到找到目标记录为止。

顺序文件的组织比较简单,向文件输出时只要把数据记录一个接一个地写入到文件中即可,但维护较为困难。为了修改某条记录,必须将整个文件读入内存,修改完后再重新写入文件。追加记录只能在文件尾进行。这种对顺序文件的读写方式称为"顺序存取"。由于顺序文件不能灵活地存取和增减数据,因而适用于有一定规律且不经常修改的数据。它的主要优点是占用空间少,容易使用。

随机文件(Random Access File):又称为直接存取文件,在随机文件中,每个记录的长度是固定的,记录中的每个字段的长度也是固定的。为了存取这类文件,需要预先明确记录的格式。随机文件中的每条记录都有一个记录号。在写入数据时,只要指定记录号,就可以把数据直接存入指定位置。而在读取数据时,也只需要给出记录号,就能直接读取该记录,而不必考虑各个记录的排列顺序或位置,可根据需要访问文件中的任一个记录。对随机文件这种按记录号进行读写的方式称为"随机存取"方式。

在随机文件中,可以同时进行读、写操作,因而能够快速地查找和修改每个记录,不必因为修改某个记录而对整个文件进行读、写操作。随机文件的优点是数据的存取灵活、修改方

便,主要缺点是占空间大,数据组织较为复杂。

(3) 按数据的编码方式分类可以将文件分为文本文件和二进制文件。

文本文件(Text File):又称 ASCII 文件,文件中的数据以文本字符(ASCII 码)进行存储。这些字符既可以是 ASCII 码字符集中的字符,也可以是汉字字符。这种文件可以用字处理软件建立和修改(必须按纯文本文件保存)。

二进制文件(Binary File):这类文件中的数据是以二进制形式(即数据在内存中的存储形式)保存的,它以字节数来定位数据,应用程序可用各种方式对其进行存取。二进制文件不能用普通的字处理软件进行编辑,但占用空间少。

8.2 文件系统操作

在 Visual Basic 中,对数据文件的操作通常按照打开(或建立)文件、读写文件及关闭文件三个步骤进行。

8.2.1 文件的打开与关闭

1.文件的打开

一个文件必须打开或建立才能使用,Visual Basic 用 Open 语句打开或建立一个文件。完整的 Open 语句的格式为:

Open 文件名[For 方式][Access 存取类型][锁定]As[#]文件号[Len=记录长度]

Open 语句的功能是:为文件的输入输出分配缓冲区,确定文件的打开方式和存取类型,以及利用所给定的"文件号"与打开的文件相关联。

说明:

(1) 格式中的 Open,For,Access,As 以及 Len 为关键字。

(2) "文件名":用来指明要打开(或建立)的文件。文件名可以带驱动器及路径说明。

(3) "方式":放在关键字 For 之后,指定文件的输入输出方式(打开方式),表 8-1 给出了文件的几种输入输出方式。

表 8-1 文件的输入输出方式

方 式	说 明
Output	指定打开文件为顺序输出方式。若指定的文件存在,则文件中原有数据将被覆盖,新的数据将从文件的开头写入;若指定的文件不存在,则创建一个新文件
Input	指定打开文件为顺序输入方式。指定的文件必须存在,否则产生错误
Append	指定打开文件为顺序输出方式。与 Output 方式不同的是,若指定文件存在,则文件中原有数据被保留,新的数据将从文件尾开始添加;若文件不存在,则创建一个新文件
Random	指定打开文件为随机存取方式,这是缺省方式。打开的文件即可以读,也可以写。当打开的文件不存在时,则创建一个新文件
Binary	指定打开文件为二进制文件。在这种方式下,可以用 Get 和 Put 语句对文件中任何字节位置的信息进行读写

(4) "存取类型":放在关键字 Access 之后,用来指定访问文件的方式,它的值可取以下

几种。

Read:打开只读文件。此时打开的文件作为输入文件,即只能从文件中读取数据,而不能向文件写入数据。

Write:打开只写文件。此时打开的文件作为输出文件,即只能向文件中写入数据,而不能从文件中读取数据。

Read Write:打开读写文件。这种类型只对随机文件、二进制文件及用 Append 方式打开的文件有效。即可以从文件中读取数据,也可以向文件写入数据。

(5)“锁定”:该子句只在多用户或多进程环境中使用,用来限制在文件打开期间其他用户或其他进程对该文件的读写方式,具体的锁定类型见表 8-2。

表 8-2　打开文件的锁定类型

类　型	说　明
缺省	在打开文件时,如不指定锁定类型,则本进程可以多次打开文件夹进行读写;在文件打开期间,其他进程不能对该文件进行读写操作
Lock Shared	任何机器上的任何进程都可以对该文件进行读写操作
Lock Read	不允许其他进程读打开的文件
Lock Write	不允许其他进程写打开的文件
Lock Read Write	不允许其他进程读、写打开的文件。即对于打开的文件,其他进程即不允许读,也不允许写

(6)“文件号”:为一整型表达式,其值在 1～511 范围内。执行 Open 语句后,打开的文件与该文件号相关联,其后对于文件的读、写等各种操作则是通过文件打开时的文件号来进行的。

(7)“记录长度”:是一个整型表达式。随机文件记录的缺省长度为 128 字节,顺序文件记录的缺省长度为 512 字节。“记录长度”的值不能超过 32 767 字节。当打开为随机存取文件时,“记录长度”表示随机存取文件中文件记录的长度。在顺序文件中,各个记录的长度不需要与“记录长度”相同,因为顺序文件各个记录的长度可以不同。因此,当打开为顺序文件时,“记录长度”则为指定的文件缓冲区的大小。

(8) 为了满足不同存取方式的需要,对同一个文件可以用几个不同的文件号打开,每个文件号有自己的一个缓冲区。对不同的访问方式,可以使用不同的缓冲区。但当使用 Output 或 Append 方式时,必须先将文件关闭,才能重新打开文件。而当使用 Input、Random 或 Binary 方式时,不必关闭文件就可以用不同的文件号打开文件。

文件的打开示例:

(1) Open "file1.dat" For Output As #1

打开(或建立)一个名为“file1.dat”的数据文件,其文件号指定为 1。若名为“file1.dat”的数据文件不存在,则创建该文件并将其打开,使记录可以写到该文件中。若文件“file1.dat”已经存在,则打开该文件,新写入的数据将覆盖原有数据,原有数据信息将丢失。

(2) Open "file2.dat" For Append As #2

若指定的数据文件“file2.dat”存在,则将其以追加方式打开,并指定文件号为 2,新写入的记录加到文件后面,原来的数据仍在文件中。若文件“file2.dat”不存在,则创建新文件。

(3) Open "file3.dat" For Input As #1

打开存在的数据文件"file3.dat",以便从文件中读取数据。若文件"file3.dat"不存在,则出错。

(4) Open "file4.dat" For Random As #1

按随机方式打开或建立一个文件,然后读出或写入定长记录。

(5) Open "file5.dat" For Binary As #1

打开或建立二进制文件,以便从文件中读出数据或从某个字节位置开始把新的数据加到文件中。

2.文件的关闭

对文件的读写等操作结束后,要及时将文件关闭。文件的关闭通过 Close 语句实现,格式为:

Close [#]文件号][,[#]文件号]...

Close 语句的作用是结束文件的输入输出操作,关闭该文件,释放文件的控制权。

例如:

Close #1

则将已打开的文件号为 1 的文件关闭。其后若想再对该文件进行操作,则必须再重新打开该文件。

说明:

(1) 格式中的"文件号"为使用 Open 语句打开的文件的文件号。

(2) 利用一个 Close 语句可以关闭多个文件。若关闭多个文件,文件号之间用","分隔。例如:

```
Close #1,#2     '将已打开的文件号为 1 和 2 的文件关闭
```

(3) 在 Close 语句中若不指定文件号,则把所有打开的文件全部关闭。此外,程序结束时将自动关闭所有打开的数据文件。例:

```
Close           '将已打开的所有文件关闭
```

(4) 数据文件使用完毕后,要及时将其关闭。关闭数据文件的用途:一是释放与该文件相联系的文件号,以供其他 Open 语句使用;二是释放打开文件所占的系统资源。另外,因为文件与内存之间的信息交换是通过文件缓冲区进行的,如果关闭的是为输出而打开的文件,则在关闭的同时将缓冲区中剩余的内容写入文件中;若不使用 Close 语句关闭文件,则在程序结束后文件将自动关闭,此时缓冲区中剩余的内容不会被写入文件中,将造成数据丢失。

8.2.2 文件管理函数与语句

为了方便用户管理文件,Visual Basic 提供了与文件(或目录)相关的操作函数或语句。下面介绍常用的文件管理函数及语句。

1.FreeFile 函数

FreeFile 函数的返回值为 Integer 类型,利用该函数可以求得一个在程序中尚未使用的文件号。用该函数取得的文件号可以避免文件号的冲突。

使用格式为:

FileNumber = FreeFile

其中 FileNumber 为 Integer 型变量,用于保存 FreeFile 函数返回的下一个可供 Open 语句使用的文件号。

2. LOC 函数

LOC 函数的使用格式为：

FileLocation = LOC(FileNumber)

其中 FileLocation 为 Long 型变量，FileNumber 为 Integer 型变量，其值为有效的文件号。

LOC 函数返回一个 Long 型数值，表示文件号为 FileNumber 文件的当前读写位置。该函数常用于随机文件和二进制文件。当用于随机文件中时，函数返回上一次对文件进行读出或写入的记录号；当函数用于二进制文件时，则返回值为上一次读出或写入的字节位置。

3. Seek 函数与语句

Seek 函数用来返回指定文件的下一个读写位置，而 Seek 语句用来设置指定文件的下一个读写位置。

(1) Seek 函数。Seek 函数的使用格式为：

Seek(FileNumber)

Seek 函数返回一个 Long 型数值，表示文件号为 FileNumber 的文件的下一个读写位置。当文件是以 Random 打开时，该函数返回的值为下一个读或写的记录号；当文件以其他方式打开时，则函数返回的值为下一个操作将要发生的字节位置。

与 Loc 函数不同的是，Seek 函数返回的是下一个要读或写的字节(或记录)位置，而 Loc 函数返回的是最近一次读或写的字节(或记录)位置。

例如，有如下程序段：

```
Open "c: \ mydir \ studata. dat" For Random As #1 Len = 32
                                        '打开记录长度为 32 字节的随机文件
Position1 = Loc(1)
Posittion2 = Seek(1)
```

程序段执行后，则 Position1 = 0(未发生任何记录的读/写)、Position2 = 1(下一个要读/写的记录号为 1)。

(2) Seek 语句。Seek 语句用来设置已打开文件的下一读写位置，其使用格式为：

Seek [#]FileNumber Position

其中，Position 为 Long 型，用来指定文件号为 FileNumber 文件的下一个要进行读写的字节或记录号(对于随机文件而言)。

4. LOF 函数

LOF 函数的使用格式为：

FileLength = LOF(FileNumber)

LOF 函数返回一个 Long 数据类型的数值，表示已打开的文件号为 FileNumber 文件的长度(即给文件分配的字节数。)

5. FileLen 函数

FileLen 函数的使用格式为：

FileLength = FileLen(FileName)

与 LOF 函数相同的是 FileLen 函数也返回一个代表一个文件长度的 Long 类型的数值(字节数)，不同的是该函数还可以求未打开文件的长度。

其中，参数 FileName 为文件名，是用来指定文件存储位置和文件名的字符串表达式。

需特别说明，如果所指定的文件已经打开，FileLen 函数返回的则是这个文件在打开前的长度。

6. EOF 函数

EOF 函数用于测试指定文件的结束状态，用 EOF 函数可以避免因试图在文件结尾处进行读操作而产生错误。其使用格式如下：

EOF(FileNumber)

其中参数 FileNumber 是一个 Integer 类型量，表示已打开的文件号。

EOF 函数返回一个 Boolean 类型值，表示文件的结束状态。当 FileNumber 所指定文件到达了文件末尾时，EOF 函数返回 TRUE，否则返回 FALSE。

EOF 函数一般用于顺序文件和随机文件。对于二进制文件一般使用 LOF 和 LOC 函数代替 EOF 函数。

7. FileAttr 函数

利用 FileAttr 函数可以求得已打开文件的输入输出方式，其使用格式为：

FileAttr(FileNumber)

FileAttr 函数返回一个枚举值，表示文件号为 FileNumber 的文件的输入输出方式，具体说明见表 8－3。

表 8－3 FileAttr 函数的返回值

返回值	输入输出方式
1	Input
2	Output
4	Random
8	Append
32	Binary

8. FileCopy 语句

利用 FileCopy 语句可以实现文件的复制，其使用格式如下：

FileCopy FileName1, FileName2

其中 FileName1 和 FileName2 为字符型表达式，FileName1 指定要复制的文件名(即源文件名)，FileName2 指定目标文件名。FileName1 和 FileName2 可以包括文件所在的驱动器和路径。注意，若使用 FileCopy 复制已打开的文件，则会发生错误。

例如：

FileCopy "c:\myfile1.dat", "d:\data\myfile2.dat"

则将 C 盘根目录下的“myfile1.dat”文件复制到“d:\data”，其名为“myfile2.dat”。

9. Name 语句

Name 语句用于实现文件名的更改，其使用格式为：

Name OldFileName As NewFileName

其中 OldFileName 和 NewFileName 为字符型表达式，分别指明原有文件名(可以带路径)和新文件名(可以带路径)。OldFileName 和 NewFileName 所指定的文件可以不在同一目录中，但必须在同一个驱动器上。

该函数的作用是将原有的 OldFileName 所指定的文件重新命名并移动到 NewFileName

所指定的文件。

10. Kill 语句

Kill 语句用于将指定的文件删除，其使用格式为：

Kill FileName

Kill 语句作用是将 FileName 所说明的文件(可以带驱动器及路径说明)删除。

Kill 语句支持通配符“*”(多字符)和“?”(单字符)的使用，从而一次可以删除多个文件。

例如：

Kill "c:\mydata\file?.dat"

表示将 C 盘 mydata 目录下文件名由 5 个字符组成，前 4 个字符为“file”，扩展名为“.dat”的所有文件删除。

Kill "c:\mydata*.dat"

表示将 C 盘 mydata 目录下所有的扩展名为“.dat”的文件删除。

11. Dir 函数

Dir 函数用于返回与指定模式及文件属性相匹配的文件名、目录名及驱动器卷标的字符串。其使用格式为：

Dir[(FileName[,Attributes])]

说明：

(1) 参数 FileName 为可选项，是字符型表达式，用来指定文件名、目录、驱动器名等。且该参数支持通配符“*”和“?”，用来指定多个文件。

(2) 参数 Attributes 为枚举类型或数值表达式，用来指定文件属性。如果省略则会返回相匹配的所有文件。表 8-4 给出参数 Attributes 的枚举值。

表 8-4　Dir 函数中参数 Attributes 的枚举值

值	常　量	说　明
Normal	vbNomal	缺省值。指定普通文件
ReadOnly	vbReadOnly	指定只读文件
Hidden	vbHidden	指定隐藏文件
System	vbSystem	指定系统文件
Volume	vbVolume	指定卷标。如果指定了任何其他属性，则忽略 vbVolume
Directory	vbDirectory	指定目录或文件夹
Archive	vbArchive	自从上次备份后文件已更改
Alias	vbAlias	文件具有不同名称

(3) 在第一次调用该函数时，参数 FileName 必须指定，返回相匹配的第一个文件名。若要得到其他相匹配的文件名，再一次调用该函数时可以不带参数。

12. MkDir 语句

MkDir 语句用来创建新的文件目录(文件夹)，其格式为：

MkDir 目录名

“目录名”中可以包含驱动器名及路径。若指定路径，则在指定路径下创建一个新的目

录,否则在当前工作驱动器的当前工作目录中创建新目录。

例如,用以下语句可以在 C 盘根目录的“mydata”目录下创建新目录“data”:

```
MkDir "C:\mydata\data"
```

13. ChDrive 语句

ChDrive 语句用于改变系统当前工作驱动器,其格式为:

```
ChDrive 驱动器名
```

例如,将当前驱动器改为 D 盘:

```
ChDrive "D"    '若“驱动器名”为字符串有多个字符,则只有第一字符有效
```

14. ChDir 语句

ChDir 语句用于改变系统工作目录,其格式为:

```
ChDir 目录名
```

例如,将 C 盘的工作目录改为“c:\mydata”:

```
ChDir "c:\mydata"
```

注意,此语句只改变指定盘的工作目录,并不能改变当前工作驱动器。

15. CurDir 语句

CurDir 语句用于返回指定驱动器的当前路径,其一般格式为:

```
PathName = CurDir(驱动器名)
```

说明:

(1) PathName 为字符串类型量。

(2) 若不指定“驱动器名”,则返回系统当前工作盘的工作目录,若驱动器名有多个字符,则只有第一个字符有效。

例如,使用以下语句可以返回 C 盘的当前工作目录:

```
PathName = CurDir("C")
```

16. RmDir 语句

RmDir 语句用于删除指定目录,其一般格式为:

```
RmDir 目录名
```

例如,将“c:\mydata”目录下的子目录“data2”删除:

```
RmDir "c:\mydata\data2"
```

RmDir 语句只能删除空目录,若指定目录中有文件,可先用 Kill 语句删除所有文件。

17. Shell 函数

在 Visual Basic 应用程序中可以利用 Shell 函数来运行一个计算机系统中的可执行程序。该函数的一般格式为:

```
Shell(FileName [,WindowStyle])
```

说明:

(1) 必选参数 FileName 用来指定要运行的可执行程序的文件名,它可以包括驱动器名和路径。

(2) 可选参数 WindowStyle 为整型量,用来指定要运行程序的窗口样式。该参数若省略,则默认值为 2,即程序以最小化窗口执行,并具有焦点。表 8-5 给出了 WindowStyle 参数可用的窗口样式值。

表 8-5　WindowStyle 参数的窗口样式值

常　量	值	说　明
vbHide	0	隐藏窗口并获得焦点
vbNormalFocus	1	以窗口最近一次设置的样式显示，并为窗口提供焦点，使该窗口为当前活动窗口
vbMinimizedFocus	2	为缺省值，以最小化的形式显示窗口，并获得焦点
vbMaximizedFocus	3	以最大化的形式显示窗口，并获得焦点
vbNormalNoFocus	4	以窗口最近一次设置的样式显示，该程序窗口不获得焦点，当前活动窗口保持不变
vbMinimizedNoFocus	6	以最小化的形式显示窗口，不获得焦点

(3) 如果该函数成功调用了指定的文件，则返回执行程序的任务 ID(任务 ID 是一个惟一的数值，用来标识正在运行的程序)，否则会产生错误(如指定文件不是可执行程序)。

例如，用以下程序可以打开系统的画图程序(文件名为"mspaint.exe")：

```
Dim ProcID As Integer
On Error Resume Next
ProcID = Shell("C:\Windows\System32\mspaint.exe", vbNormalFocus)
                                    '运行画图程序，并以正常窗口显示
If Err.Number > 0 Then    '当有错误发生时(如"mspaint.exe"不存在)
  MsgBox "画图程序不存在或路径错!", vbCritical
End If
```

8.3　顺序文件

顺序文件为普通的文本文件。这种文件的组织结构比较简单，文件中的记录按顺序存放，记录是以换行符为分隔符，一行一条记录，记录可长可短。在顺序文件中，记录的逻辑顺序与存储顺序相一致，对文件的读写操作只能采用"顺序存取"方式，即只能按顺序读写一行。顺序文件一般用来存储字符、数字以及其他可用 ASCII 码表示的数据。由于"顺序存取"特点，顺序文件一般适用于存放有一定规律且不经常修改的数据。

8.3.1　顺序文件的打开与关闭

顺序文件的打开与关闭仍使用 Open 语句和 Close 语句实现。

1.顺序文件的打开

使用 Open 语句打开顺序文件的格式为：

Open 文件说明 [For 方式] [锁定] As [#]文件号 [Len=缓冲区大小]

说明：

(1) 关键字 For 后的文件输入输出方式只能是 Output(写方式)、Input(读方式)以及 Append(追加方式)中的一种。

(2) 当使用 Open 语句打开一顺序文件时，"Len=缓冲区大小"子句用来指定打开文件的缓冲区的大小(以字节为单位)，其缺省值为 512 字节。

例如：

```
Open "C:\mydir\mydata1.txt" For Output As #1
```

表示以写方式打开(或创建)指定文件,文件缓冲区默认长度为 512 字节。

```
Open "C:\mydir\mydata2.txt" For Input As #2 Len=256
```

表示以读方式打开指定文件,并指定文件缓冲区长度为 256 字节。

```
Open "C:\mydir\mydata3.txt" For Append As #3
```

表示以追加方式打开指定文件,可以向文件尾追加数据,原有数据不丢失。

2.顺序文件的关闭

当对文件的操作结束后,一定要及时使用 Close 语句将其关闭,特别是以 Output 和 Append 方式打开的文件,否则有可能造成数据的丢失。

例如:

```
Close #1          '关闭文件号为 1 的文件
Close #2,#3       '关闭文件号为 2 和 3 的文件
Close             '关闭所有的已打开的文件
```

8.3.2 顺序文件的读写操作

文件正确打开后,即可利用打开时指定的文件号对其进行读和写操作。对于顺序文件,Visual Basic 提供的写操作语句有 Print # 语句和 Write # 语句,读操作语句有 Input # 语句和 Line Input # 语句。

1.顺序文件的写操作

当顺序文件以 Output 或 Append 方式打开后,即可利用 Print # 语句和 Write # 语句向文件中写入数据。

(1) Print # 语句。Print # 语句的使用格式为:

```
Print #文件号,[输出列表]
```

Print # 语句的功能是按指定格式将数据写入"文件号"所指定的文件中。

说明:

① 输出列表为用分号或逗号分隔的变量、常量、空格和定位函数序列等。

② Print # 语句中的输出列表可以省略,这时将向文件中写入一个空行。

例如:

```
Print #1,
```

表示向文件号为 1 的文件中写入一空行,其后的","不能省略。

③ Print # 语句中的各数据项之间可以用";"分隔,也可以用","分隔,分别对应紧凑格式和标准格式。当使用";"作为分隔符输出数值数据时,因为数值数据前有符号位,后有空格,因此不会给以后读取造成麻烦;而当利用";"作为分隔符输出字符串数据时,因为输出的字符串数据之间没有空格,则可能会引起麻烦。

例如,如下程序段:

```
Open "c:\test1.txt" For Output As #1
a$ ="Apple": b$ ="Banana": c$ ="Orange"
Print #1, a$, b$, c$          '用逗号分隔
Print #1, a$; b$; c$          '用分号分隔
Close #1
```

程序段执行后,写入到文件"test1.txt"中的信息为:

Apple　　Banana　　Orange

AppleBananaOrange

使用";"作为分隔符时,为了使输出的各字符串分开(字符串能够被正确读出),可以在输出各个字符串之间人为地插入逗号进行分隔。例如,改用如下语句:

```
Print #1, a$; ",";b$;",";c$
```

则该语句写入到文件中的信息为:

```
Apple,Banana,Orange
```

若字符串本身包含逗号、分号、空格及回车等符号,则须人为插入双引号""(ASCII 码为 34)进行分隔。例如:

```
a$ ="Student Name":b$ ="Birthday(mm,dd,yy)"
Print #1, Chr(34);a$ ;Chr(34); Chr(34);b$ ; Chr(34)
```

则写入到此文件中的信息为:

```
"Student Name" "Birthday(mm,dd,yy)"
```

④ 在实际上,Print # 语句仅仅是将数据输出到缓冲区,而数据由缓冲区写到文件中的操作是由文件系统来完成的。只有在缓冲区已满、缓冲区未满但开始执行下一个 Print # 语句以及关闭文件时,才将缓冲区中的数据写入文件中。

(2) Write # 语句。Write # 语句的使用格式为:

```
Write #文件号,[输出列表]
```

Write # 语句的功能与 Print # 语句基本相同,将指定数据写入"文件号"所指定的文件中。Write # 语句各个输出项之前一般用",分隔。与 Print #语句不同的是使用 Write # 语句向文件写入数据时,数据在磁盘上总是以紧凑格式存放,数据项之间自动插入","分隔,并且对于字符串自动加上双引号。

例如,如下程序段:

```
Open "c:\vbtext2.txt" For Output As #1
StuName$ ="Li Ming": Birthday$ ="03,15,88"
Score=87
Write #1, StuName$ , Birthday$ , Score
Close #1
```

程序段执行后,写入到文件"test1.txt"中的信息为:

```
"Li Ming","03,15,88",87
```

2.顺序文件的读操作

当顺序文件以 Input 方式打开后,即可利用 Input # 语句、Line Input # 语句及 Input 函数从文件中读取数据。

(1) Input # 语句。Input # 语句的使用格式为:

```
Input #文件号,变量表
```

其功能是从由"文件号"所指定的文件中,读取数据赋给变量表中的各个变量。

例如:

```
Dim num1 As Integer, num2 As Integer
Dim str1 As String, str2 As String
```

Input # 1, num1, num2, str1, str2

则表示从已打开的文件号为 1 的文件中读取两个整数分别赋给变量 num1 和 num2,以及读取两个字符串分别赋给字符串变量 str1 和 str2。

说明:

① 格式中的"变量表"由一个或多个变量组成,多个变量之间用逗号分隔。"变量表"中变量的类型和次序必须与文件中对应的数据项相匹配,变量可以是不同类型的,但不能为结构类型。

② 利用 Input # 语句从文件中读取数值数据时,将忽略前导空格、回车和换行符,并且空格、回车和换行符可以作为数值数据的分隔符。

③ 利用 Input # 语句从文件中读取字符串数据时,若有双引号则以双引号作为字符串的标志,否则以逗号或行结束符作为分隔符,空行为零长度的字符串。另外,当以逗号及行结束符作为字符串的分隔时,将忽略前导及后缀空格符。

(2) Line Input # 语句。Line Input # 语句的格式为:

Line Input # 文件号,字符串变量

Line Input # 语句的作用是从给定"文件号"的文件中读取一个完整的行,并将它赋给一个"字符串变量"。

例如:

Line Input # 1,TextLine

则表示从文件号为 1 的文件中读取一行字符,赋给字符串变量 TextLine。

(3) Input 函数。利用 Input 函数也可以从顺序文件中读取字符串,其使用格式为:

字符串变量 = Input(长度,文件号)

Input 函数的作用是从由"文件号"所指定的文件中读取指定长度的字符串并作为函数值返回。其中空格、逗号、换行符及回车符等都是有效字符。

例如,要从一个文件号为 FileNumer 的顺序文件中读取 20 个字符并赋给字符串变量 Str1,则为:

Str1 = Input(20,FileNumber)

8.3.3 顺序文件应用实例

下面通过实例来说明顺序文件的使用方法。

【例 8-1】 利用 Input # 和 Write # 语句读写顺序文件,设计"我的通讯录"管理程序。

该程序能够完成通讯录的浏览、添加等功能。在实例中各控件的主要属性见表 8-6,各控件的布局及程序运行界面如图 8-1 所示。

表 8-6 实例中用到的控件及主要属性设置表

对 象	属 性	属性值	说 明
Form1	Caption	我的通讯录	Form 控件
Frame1	Caption	(空)	Frame 控件
Label1	Caption	姓名:	Label 控件
Label2	Caption	电话:	Label 控件
Label3	Caption	Email:	Label 控件

续表 8-6

对　象	属　性	属性值	说　明
TextName	Text	(空)	Text 控件,用于显示或录入"姓名"
TextTel	Text	(空)	Text 控件,用于显示或录入"电话"
TextEmail	Text	(空)	Text 控件,用于显示或录入"Email"
Frame2	Caption	浏览	Frame 控件
Text1	Text	(空)	Text 控件,用于浏览通讯录信息
	Locked	True	
	MultiLine	True	
Frame3	Caption	(空)	Frame 控件
CmdAppend	Caption	添加	Command 控件
CmdSave	Caption	保存	Command 控件
CmdOpen	Caption	打开	Command 控件
CmdExit	Caption	退出	Command 控件

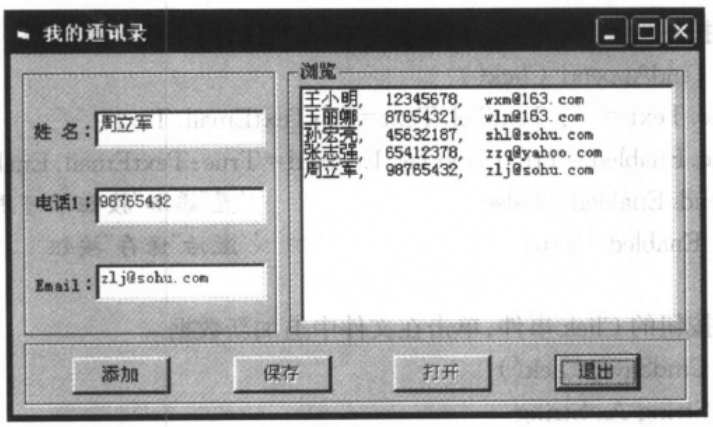


图 8-1　"我的通讯录"运行界面

程序中的代码编写如下:

(1) 窗体的 Load 事件,设置控件的 Enabled 属性。

```
Private Sub Form_Load()
   TextName.Enabled = False: TextTel.Enabled = False: TextEmail.Enabled = False
   CmdSave.Enabled = False: CmdAppend.Enabled = False
   CmdOpen.Enabled = True: CmdExit.Enablled = True
End Sub
```

(2) "打开"按钮的 Click 事件,单击该按钮则在 List 控件中显示通讯录信息。

```
Private Sub CmdOpen_Click()
   Dim Name As String, TelNum1 As String, Email As String
   Dim TextString As String
   On Error Resume Next
Open "C:\TelBook.txt" For Input As #1        '以输入方式打开通讯录文件
```

```
  If Err.Number > 0 Then                    '若文件不存在时,给出提示
    MsgBox "文件不存在,请先创建文件!", vbOKOnly, "提示"
  Else
    TextString = ""
    Do While Not EOF(1)     '利用循环读出顺序文件中的所有数据
      Input #1, Name, TelNum, Email
      TextString = TextString & Name & "," & TelNum & ","& Email _
        & Chr(13) & Chr(10)
    Loop
    Close #1
    Text1.Text = TextString                 '在 Text1 中显示读出的数据信息
  End If
  CmdOpen.Enabled = False                   '置"打开"按钮不可用
  CmdAppend.Enabled = True                  '激活"添加"按钮
End Sub
```

(2) "添加"按钮的 Click 事件,单击激活 Text 控件,用于录入新数据。

```
Private Sub CmdAppend_Click()
  TextName.Text = "": TextTel.Text = "":TextEmail.Text = ""
  TextName.Enabled = True:TextTel.Enabled = True:TextEmail.Enabled = True
  CmdAppend.Enabled = False                 '置"添加"按钮不可用
  CmdSave.Enabled = True                    '激活"保存"按钮
End Sub
```

(3) "保存"按钮的 Click 事件,单击在文件中追加新数据。

```
Private Sub CmdSave_Click()
  Dim TextString As String
  TextString = ""
  Open "c:\TelBook.txt" For Append As #1    '以追加方式打开通迅录文件
  Write #1, TextName.Text, TextTel.Text, TextEmail.Text  '将新数据写入文件
  Close #1
  TextString = TextName.Text & "," & TextTel.Text & "," & TextEmail.Text & Chr _
  (13) & Chr(10)
  Text1.Text = Text1.Text & TextString      '在 Text1 中显示添加的新数据
  TextName.Enabled = False:TextTel.Enabled = False: TextEmail.Enabled = False
  CmdAppend.Enabled = True: CmdSave.Enabled = False
End Sub
```

(4) "退出"按钮的 Click 事件,实现退出程序功能。

```
Private Sub CmdExit_Click()
  Unload Me
End Sub
```

8.4　随机文件

在随机文件(有时又称为记录文件)中,每个记录的长度是固定的,记录中的每个字段的长度也是固定的。在这种文件结构中,每个记录都有其惟一的记录号,所以在读取数据时,只要知道记录号,便可以直接读取记录。这种存取方式称为"随机存取"。

随机文件的记录可以由一个或多个字段组成,只由一个字段组成的记录对应于任一标准类型,具有多个字段的记录对应于用户自定义类型(记录类型)。

例如,描述一名学生信息的记录类型 StuType 如下:

```
Type StuType
  ID As String * 8              '数据项(字段)ID 用来表示学生的学号
  Name As String * 20           '数据项 Name 用来表示学生的姓名
  Age As Integer                '数据项 Age 用来表示学生的年龄
  Score As Integer              '数据项 Score 用来表示学生的成绩
End Type
```

用户自定义的记录类型量所占的存储空间(记录长度)为每个字段的长度(所占字节数)之和。如以上定义的 StuType 类型量的长度为:

8(ID 的长度) + 20(Name 的长度) + 2(Age 的长度) + 2(Score 的长度) = 32

定义完成 StuType 记录类型后,即可用其定义一个变量来存储学生的信息。如用其定义一个 StuType 类型的变量 Student1 如下:

```
Dim Student1 As StuType
```

下面就以上面自定义的 StuType 类型为例,介绍随机文件的操作。

8.4.1　随机文件的打开与关闭

随机文件的打开与关闭仍使用 Open 语句和 Close 语句来实现,只是在打开随机文件时指定的输入输出方式为 Random 方式。

1.随机文件的打开

利用 Open 语句打开随机文件的一般格式为:

```
Open 文件说明 For Random As [#] 文件号 Len = [记录长度]
```

其中,For 子句所指定的"Random"即表明打开的文件为随机文件。Len 子句中的"记录长度"表示该文件中每个记录所包含的字节数。

利用此格式打开的随机文件,即可对其进行读操作,也可对其进行写操作。并且当打开的文件不存在时,则创建一个新的文件。

例如,打开一个记录类型为 StuType 类型的随机文件"C: \ mydir \ studata. txt"的 Open 语句如下:

```
Open "c: \ mydir \ studata. dat" For Random As #1 Len = 32
```

2.随机文件的关闭

使用 Close 语句关闭随机文件的方法与关闭顺序文件相同。如关闭上面打开的随机文件,使用的语句如下:

```
Close #1
```

8.4.2 随机文件的读写操作

Visual Basic 提供 Put # 语句和 Get # 语句分别用来实现对随机文件的读写操作。

1.随机文件的写操作

随机文件的写操作语句 Put # 的使用格式为：

Put #文件号,[记录号],变量

Put # 语句的作用是将“变量”的内容写到“文件号”所指定的文件中去,写入的具体位置由“记录号”指定。

说明：

(1) 格式中的“记录号”指明要将变量内容写入到哪个记录中,若省略,则在上次调用 Put # 语句或 Get # 语句(读操作语句)写或读过记录的下一条记录或最近一次由 Seek 语句定位的记录中写入。若没有执行过 Get # 语句、Put #语句以及 Seek 语句,则从第 1 条记录开始写。

(2) 格式中的“变量”可以为任意类型,一般为用户自定义的记录类型。若文件中记录的结构是单一的标准类型,此处还可以是相应的表达式。

例如,如下程序段实现向随机文件中写入两个学生的数据信息。

```
Dim student1 As StuType ,student2 As StuType
student1.ID="08060301" : student1.Name="Li Ming"
student1.Age=18: student1.Score=86
student2.ID="08060302" : student2.Name="Wang Na"
student2.Age=19: student2.Score=79
Open "stuinfo.dat" For Random As #1 Len=Len(student1)
                                    'Len 函数用来求记录类型量的长度
Put #1,1,student1                   '将 student1 的内容写入第 1 个记录
Put #1, , student2                  '将 student2 的内容写入下一个(第 2 个)记录
Close #1
```

2.随机文件的读操作

使用 Get # 语句可以从随机文件中将指定记录的数据读出存入相应的变量中。其使用格式为：

Get # 文件号,[记录号],变量

说明：

(1) 格式中的“记录号”指定了要将哪个记录的数据读入到变量中,若省略,则将上次调用 Put # 语句或 Get # 语句(读操作语句)写或读过记录的下一条记录或最近一次由 Seek 语句定位记录的数据内容读出。若没有执行 Get # 语句、Put #语句以及 Seek 语句,则从第 1 条记录开始读。

(2) 格式中的“变量”类型(或结构)应与文件中记录的结构相一致。

例如,可以利用以下程序段实现从学生数据信息文件“stuinfo.dat”中读取第二个学生的数据。

```
Dim student1 As StuType
Open "stuinfo.dat" For Random As #1 Len=Len(student1)
```

```
Get #1，2，student1          '将第2个记录的内容读入student1中
Close #1
```

8.4.3　随机文件记录的维护

利用随机文件的读写操作语句，可以实现随机文件记录的修改、添加及删除等维护操作。

1. 随机文件记录的修改

要修改随机文件中的某条记录，可以利用 Put # 语句将原有记录信息替换（重新写入新内容）的方法完成。

例如，利用如下语句即可将指定文件中第2条记录的内容修改为 student1 的数据内容：

```
Put #1, 2 ,student1
```

2. 随机文件记录的添加

若要在随机文件的尾端添加新记录，可以利用 Put # 语句在“文件的记录数加1”的位置写入新记录数据即可。

如设当前打开的文件号为1的文件“studata.dat”中存储的记录数为2，则利用下面的语句可以向文件尾端追加新记录：

```
Put #1, 3 , student1          '记录号为文件中原有记录数加1
```

然而，往往文件中的记录数并不知道，此时可以利用 Lof 函数和记录长度取得当前文件中的记录数。因此以上语句可改为：

```
RecordNum = Lof(1)/Len(student1)          '求打开文件中的已有记录数
Put #1, RecordNum + 1 , student1
```

利用 Get # 语句和 Put # 语句还可以实现向随机文件中的中间位置插入一条新记录，读者可自行思考。

3. 随机文件记录的删除

可以通过以下几个步骤实现对随机文件记录的删除：

(1) 创建并打开一个新的临时文件。

(2) 打开欲删除记录的文件，并将所有有用的记录复制到新建临时文件中。即按顺序将原有文件中有用的记录读出，并写入到新建临时文件中。

(3) 关闭这两个文件，并使用 Kill 语句将原有文件删除。

(4) 使用 Name 语句将新建临时文件以原有文件的文件名重新命名。

8.4.4　随机文件应用实例

下面通过一个实例说明随机文件的使用。

【例8-2】 利用随机文件实现一个学生档案管理程序，能够实现学生信息（包括学号、姓名、年龄、性别及成绩）的录入、修改、删除及浏览等功能。

图8-2给出了本实例的程序运行界面及各控件的布局。实例中用到的控件及主要属性见表8-7。

程序中的代码编写如下：

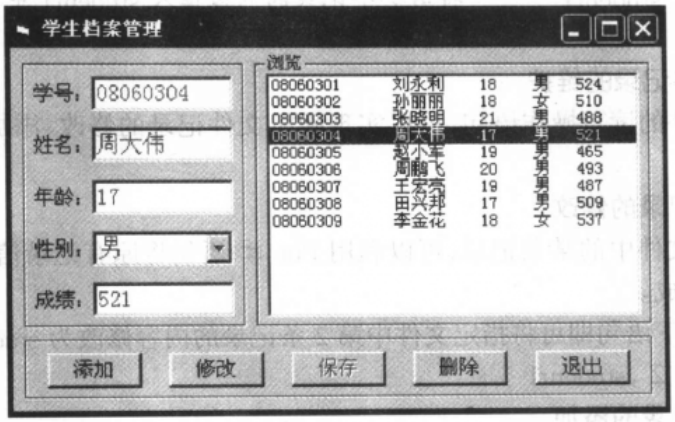


图 8－2 “学生档案管理”程序运行界面

表 8－7 “学生档案管理程序”的控件及主要属性设置表

对 象	属 性	属性值	说 明
Form1	Caption	学生档案管理	Form 控件
Frame1	Caption	(空)	Frame 控件
Label1	Caption	学号	Label 控件
Label2	Caption	姓名	Label 控件
Label3	Caption	年龄	Label 控件
Label4	Caption	性别	Label 控件
Label5	Caption	成绩	Label 控件
Text1(0)	Text	(空)	Text 控件,用于显示或录入“学号”
Text1(1)	Text	(空)	Text 控件,用于显示或录入“姓名”
Text1(2)	Text	(空)	Text 控件,用于显示或录入“年龄“
Text1(3)	Text	(空)	Text 控件,用于显示或录入“成绩
Combo1	Text	(空)	ComboBox 控件,用于显示或录入“性别”
Frame2	Caption	浏览	Frame 控件
List1	List	(空)	ListBox 控件,用于浏览学生信息
Frame3	Caption	(空)	Frame 控件
CmdAppend	Caption	添加	CommandButton 控件
CmdModify	Caption	修改	CommandButton 控件
CmdSave	Caption	保存	CommandButton 控件
CmdDel	Caption	删除	CommandButton 控件
CmdExit	Caption	退出	CommandButton 控件

(1) 在通用过程段内定义如下类型及变量。

```
Private Type StuType                    '学生记录类型
  ID As String * 8: Name As String * 8
  age As Integer: Sex As String * 2
```

```
    score As Integer
End Type
Dim Stu As StuType              '记录类型变量,用于文件记录的读写
Dim CurRecNum As Integer        '用于文件保存时,存放写入位置(记录号)
```

(2) 窗体的 Load 事件,实现数据的浏览功能

```
Private Sub Form_Load()
  Dim ID As String, Name As String, Sex As String, ItemString As String
  Dim age As Integer, score As Integer, LastRec As Integer
  List1.Enabled = True: List1.Clear          '置列表框可用并清空
  Open App.Path & "\stuinfo2.dat" For Random As #1 Len = Len(Stu)   '打开数据文件
  LastRec = LOF(1) / Len(Stu)                '求文件中记录个数
  For n = 1 To LastRec                       '读出所有记录,并加载到列表框中显示
    Get #1, n, Stu
    With Stu
      ID = Format(.ID, "C@@@@@@@@@@")
      Name = Format(Mid(.Name, 1, 4), "@@@@@@@@@@")
      age = Format(.age, "###")
      Sex = Format(Mid(.Sex, 1, 1), "@@")
      score = Format(.score, "####")
      ItemString = ID & " " & Name & " " & age & " " & Sex & " " & score
    End With
    List1.AddItem ItemString
  Next
  Close #1
  Combo1.List(0) = "男": Combo1.List(1) = "女"          '添加性别选项
  CmdSave.Enabled = False                    '设置"保存"按钮不可用
  For i = 0 To 3
    Text1(i).Enabled = False                 '设置文本框不可用,只有显示功能
  Next i
  Combo1.Enabled = False                     '设置组合框不可用,只有显示功能
  CmdAppend.Enabled = True                   '设置"添加"按钮可用
  If List1.ListCount > 0 Then                '当列表框中有学生记录时
    CmdModify.Enabled = True                 '设置"修改"按钮可用
    CmdDel.Enabled = True                    '设置"删除"按钮可用
    List1.ListIndex = 0                      '选定列表框第 1 个记录
    List1_Click     '调用 List1_Click 事件在文本框和组合框中显示选定记录
  Else              '列表框无记录,置"修改"、"删除"按钮不可用,组合框、文本框清空
    CmdModify.Enabled = False: CmdDel.Enabled = False
    For i = 0 To 3
```

```
        Text1(i).Text=""
      Next i
      Combo1.Text=""
    End If
End Sub
```

(3)“添加”按钮的 Click 事件,实现新记录的录入功能。

```
Private Sub CmdAppend_Click()
  CmdSave.Enabled=True          '“保存”按钮可用
  CmdAppend.Enabled=False       '“添加”按钮不可用
  CmdModify.Enabled=False       '“修改”按钮不可用
  CmdDel.Enabled=False          '“删除”按钮不可用
  List1.Enabled=False           '列表框不可用
  For i=0 To 3
    Text1(i).Enabled=True: Text1(i).Text="" '设置文本框可用并置空
  Next i
  Combo1.Enabled=True                     '组合框可用
  Combo1.Text=Combo1.List(0)              '组合框置初值“男”
  Text1(0).SetFocus
  CurRecNum=List1.ListCount+1             '添加记录的位置(记录号)
End Sub
```

(4)“修改”按钮的 Click 事件,实现记录的修改功能。

```
Private Sub CmdModify_Click()
  CmdSave.Enabled=True : CmdModify.Enabled=False
  CmdAppend.Enabled=False:CmdDel.Enabled=False
  List1.Enabled=False
  For i=0 To 3
    Text1(i).Enabled=True
  Next i
  Combo1.Enabled=True
  CurRecNum=List1.ListIndex+1             '要修改的记录的位置(记录号)
  Text1(0).SetFocus
End Sub
```

(5)“保存”按钮的 Click 事件,保存添加或修改的记录。

```
Private Sub CmdSave_Click()
  Dim SaveFlag As Integer
  SaveFlag=MsgBox("是否要保存文件?", vbOKCancel + vbQuestion, "提示")
  If SaveFlag=1 Then                      '确定要进行保存操作时
    With Stu
      .ID=Text1(0).Text: .Name=Text1(1).Text
```

```
        .age = Val(Text1(2).Text): .Sex = Combo1.Text
        .score = Val(Text1(3).Text)
    End With
    Open App.Path & "\stuinfo2.dat" For Random As #1 Len = Len(Stu)
    Put #1, CurRecNum, Stu          '新数据写入文件
    Close #1
  End If
  Call Form_Load                    '调用 Form_Load 事件重新显示保存后文件内容
  CmdSave.Enabled = False
End Sub
```

(6) "删除"按钮的 Click 事件,实现记录的删除功能。

```
Private Sub CmdDel_Click()
  Dim LastRec As Integer, ReNum As Integer,DelFlag As Integer
  DelFlag = MsgBox("是否要删除此记录?", vbOKCancel + vbQuestion, "提示")
  If DelFlag = 1 Then               '确定要进行删除操作时
    recnum = List1.ListIndex + 1    '获取要删除记录的记录号
    Open App.Path & "\temp.dat" For Random As #1 Len = Len(Stu)
                                                    '打开临时文件
    Open App.Path & "\stuinfo2.dat" For Random As #2 Len = Len(Stu)
                                                    '打开学生信息数据文件
    LastRec = LOF(2) / Len(Stu)     '求出当前中的记录个数
    For i = 1 To LastRec            '将有用的记录复制到临时文件中
      If i <> recnum Then
        Get #2, i, Stu: Put #1, , Stu
      End If
    Next i
    Close #1, #2
    Kill App.Path & "\stuinfo2.dat"  '删除原来的文件
    Name App.Path & "\temp.dat" As App.Path & "\stuinfo2.dat"
                                    '临时文件更名为原来文件
    Call Form_Load  '调用 Form_Load 事件重新显示删除后的文件内容
  End If
End Sub
```

(7) "退出"按钮的 Click 事件,实现退出程序功能。

```
Private Sub CmdExit_Click()
  Unload Me
End Sub
```

(8) 列表框的 Click 事件,实现在文本框和组合框中显示选定记录功能。

```
Private Sub List1_Click()
```

```
    If List1.ListIndex > -1 Then          '有记录项被选定时
      CmdModify.Enabled = True : CmdDel.Enabled = True
      Open App.Path & "\stuinfo2.dat" For Random As #2 Len = Len(Stu)
      Get #2, List1.ListIndex + 1, Stu
      Close #2
      With Stu                            '在文本框及组合框中显示选定记录项
        Text1(0).Text = .ID: Text1(1).Text = .Name :Text1(2).Text = .age
        Text1(3).Text = .score: Combo1.Text = .Sex
      End With
    End If
  End Sub
```

8.5 二进制文件

二进制文件中的数据是以二进制形式(数据在内存中的存储形式)保存的,此类文件可以看作是字节的顺序排列。可以从二进制文件中的任何一个字节处开始读写,因此文件的访问方式具有最大的灵活性。二进制文件与随机文件不同的是随机文件的存取单位是记录,而二进制文件的存取单位是字节。也可以将二进制文件看作记录长度为 1 个字节的随机文件。

8.5.1 二进制文件的操作

1.二进制文件的打开与关闭

使用 Open 语句打开二进制文件的语句格式为:

Open 文件名 For Binary As [#] 文件号

例如:

```
Open "c:\mydata\test.dat" For Binary As #1
```

执行此语句后,则"test.dat"作为二进制文件打开。如果该文件存在则将其打开,并且文件中原有数据不会丢失。如果该文件不存在,则创建新文件并将其打开。

二进制文件的关闭与其他文件的关闭相同,都使用 Close 语句来实现。

2.二制文件的读写操作

与随机文件相同,可以使用 Get # 语句和 Put # 语句来实现二进制文件的读写操作,其使用格式为:

(1) 二进制文件的写操作语句

Put # 文件号,[读写位置],变量

(2) 二进制文件的读操作语句

Get # 文件号,[读写位置],变量

与随机文件不同的是,Put # 和 Get # 语句中的"读写位置"不是记录号,而是要进行读写操作的字节位置。可以从"读写位置"指定的字节开始读取或写入多个字节,写入或读出的字节数由语句格式中"变量"的类型决定。

8.5.2　二进制文件应用实例

下面以一个实例说明二进制文件的使用。

【例 8-3】　利用二进制文件的读写操作实现文件的复制。

本实例的功能：利用两个文本框分别输入源文件和目标文件的路径及文件名，单击“复制”按钮，实现将源文件复制为目标文件。

本例中各控件的布局及程序运行界面如图 8-3 所示，各控件的主要属性设置见表 8-8。

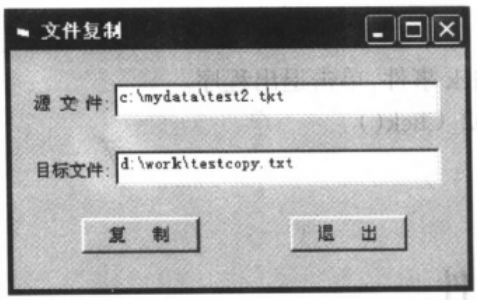


图 8-3　“文件复制”程序界面

表 8-8　文件复制程序的各控件及主要属性设置表

对　象	属　性	属性值	说　明
Form1	Caption	文件复制	Form 控件
Label1	Caption	源 文 件	Label 控件
Label2	Caption	目标文件	Label 控件
Text1	Text	(空串)	TextBox 控件，用于输入源文件名
Text2	Text	(空串)	TextBox 控件，用于输入目标文件名
Command1	Caption	复　制	CommandButton 控件
Command2	Caption	退　出	CommandButton 控件

程序中的代码编写如下：

(1) “复制”按钮的 Click 事件，实现文件复制功能。

```
Private Sub Command1_Click()
  Dim char As Byte
  On Error Resume Next
  Open Trim(Text1.Text) For Binary As #1      '打开 Text1 指定的源文件
  If Err.Number > 0 Then                      '打开源文件出错时，给出提示
    MsgBox "未指定源文件或路径错!", vbCritical, "提示"
  Else
    On Error Resume Next
    Open Trim(Text2.Text) For Binary As #2    '打开 Text2 指定的目标文件
  If Err.Number > 0 Then                      '打开目标文件出错时，给出提示
    MsgBox "未指定目标文件或路径错!", vbCritical, "提示"
  Else
    Do While Not EOF(1)
```

```
        Get #1, , char          '从源文件中读取一个字节
        Put #2, , char          '将读取的一个字节写入目标文件
      Loop
      Close
      MsgBox "文件复制完成!",,"提示"
    End If
    End If
End Sub
```

(2) "退出"按钮的 Click 事件,单击退出程序。

```
Private Sub Command2_Click()
  Unload Me
End Sub
```

8.6 文件系统控件

为了方便用户对计算机文件的管理,Visual Basic 还提供了三个文件系统控件来实现对磁盘文件目录及文件的操作。在 Visual Basic 6.0 的标准工具箱中可以找到这三个控件(图 8-4),这三个控件分别为 DriveListBox(驱动器列表框)控件、DirListBox(目录列表框)控件以及 FileListBox(文件列表框)控件。利用这三个控件的属性实现对文件的操作,这三个控件可以单独使用,也可以组合起来使用。

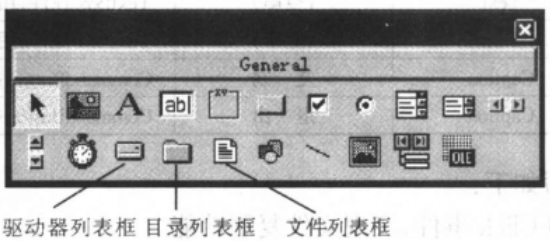


图 8-4 文件系统控件

8.6.1 DriveListBox 控件

DriveListBox(驱动器列表框)控件是一个下拉式列表框(图 8-5),只不过该下拉列表框提供的是一个驱动器列表。在默认状态下,在驱动器列表中显示的是当前驱动器。当该控件获得焦点时,用户可输入任何有效的驱动器标识符,或者单击列表框右侧的箭头(此时将显示所有有效的驱动器列表),从中选定新驱动器。

Drive 属性是 DriveListBox 控件的一个重要属性。通过读取或修改 Drive 属性的值,可以获取或修改在 DriveListBox 控件中显示的驱动器。例如,设 DriveListBox 控件名为"Drive1",则用以下语句可以将该控件的下拉列表中显示内容改为"d:":

```
Drive1.Drive="d:"
```

当然也可以通过单击列表框右侧的箭头,通过选定所列举的驱动器来改变 Drive 属性的值。

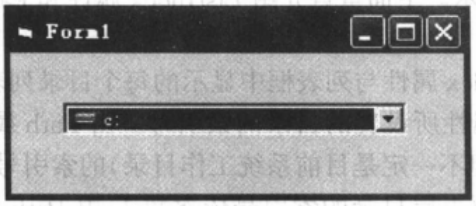


图 8－5　DriveListBox 控件

必须注意，通过改变 Drive 属性的值，只能改变在控件中显示的内容，并不能改变当前的工作驱动器。若要利用 Drive 属性的值改变当前工作驱动器，可以利用 ChDrive 语句实现。例如：

```
ChDrive Drive1.Drive
```

Change 事件则是 DriveListBox 控件的一个重要事件，当 Drive 属性值改变时则会触发该事件。如若在改变 Drive 属性值的同时改变当前工作驱动器，可以将上面给出的 ChDrive 语句添加在 Change 事件中。

8.6.2　DirListBox 控件

DirListBox（目录列表框）控件是用来实现文件目录管理的一个有效控件，目录列表框从最上层目录开始显示指定驱动器（默认情况下为当前工作驱动器）上的目录结构，该目录结构采用按层次关系缩进的结构形式。根据目录出现在列表框的最上端，指定的目录（默认情况下为系统当前工作目录）被突出显示。如图 8－6 所示。

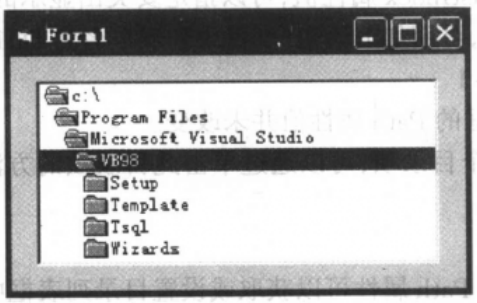


图 8－6　DirListBox 控件

DirListBox 控件的主要属性和事件见表 8－8。

表 8－8　DirListBox 控件的主要属性和事件

属性或事件	说　明	备　注
Path	获取或设置运行时选择的路径。默认路径为系统当前工作路径。该属性值的改变将触发 Change 事件	属性
ListIndex	获取或设置控件中当前被选择的目录项目的索引号	属性
ListCount	其值为 Path 属性所指定目录的所有子目录数。ListCount 值等于最大索引号加 1	属性
Change	当 Path 属性值改变时触发该事件	事件

通过获取或设置 DirListBox 控件的属性（或事件），可以获取控件所显示内容的信息，也

可以控制控件中内容的显示。下面重点介绍 ListIndex 属性和 Path 属性。

1. ListIndex 属性

目录列表框的 ListIndex 属性与列表框中显示的每个目录列表项相关联，默认情况下其值为 -1，即等于由 Path 属性所设置的目录的索引号。由 Path 属性所设置的目录(即目录列表框中的“当前目录”，但不一定是目前系统工作目录)的索引号总为 -1，而它父层目录的索引号为 -2，依次类推，再上层目录的索引号依次减 1；由 Path 属性所设置的目录第一个子目录的索引号为 0，第二个子目录的索引号为 1，再向下的子目录的索引号依次加 1。如图 8-7 所示。

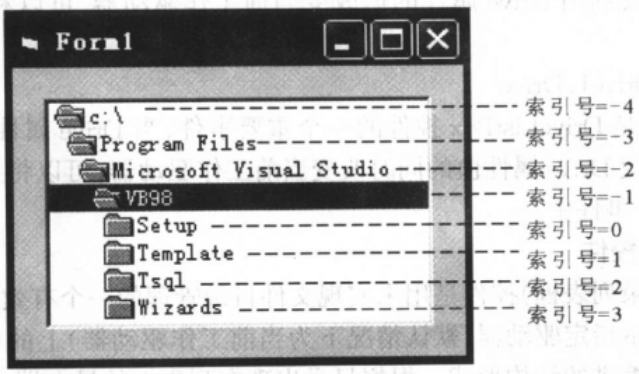


图 8-7 目录结构列表框中目录项索引值

修改目录列表框的 ListIndex 属性值，可以指定要突出显示的目录项。如在上图中若要突出显示“Template”目录，可用如下语句(设目录列表框名为 Dir1)：

```
Dir1.ListIndex=1
```

注意，此时目录列表框的 Path 属性值并未改变。

当然，要突出显示某个目录项，可以通过单击此目录项的方法实现。此时 ListIndex 属性的值亦发生相应改变。

2. Path 属性

用 DirListBox 控件的 Path 属性可以获取或设置目录列表框中“当前目录”(并不一定是系统当前的工作目录)。例如，对图 8-6 所示的目录列表框，用如下语句修改 Path 属性值(设该控件名为 Dir1)：

```
Dir1.Path="C:\Program Files\Microsoft Visual Studio\VB98\Template"
```

执行此语句后，则目录列表框显示内容如图 8-8 所示。此时“Template”目录被设置为列表框所显示的目录结构中的“当前目录”，其索引号设为 -1(同时将 ListIndex 属性值设置为 -1)。在列表框中显示“Template”目录的所有上级目录及其所有子目录。

当然，也可以通过双击某个目录项的方式来改变目录列表框中的“当前目录”以及 Path 属性的值。

在前面已多次提到目录列表框中的“当前目录”并不一定是系统当前工作目录。因此，以上通过给 Path 属性赋值或通过双击的方法修改 Path 属性值来改变列表框中所显示的目录结构，并不能改变系统当前系统工作目录。可以利用 ChDir 语句将 Path 属性所指定的目录设为当前工作目录(但不能改变当前工作驱动器，参见 8.2 节 ChDir 语句)，语句如下：

ChDir Dir1.Path

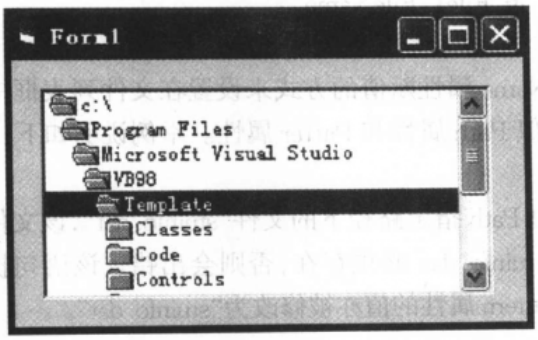


图 8-8　利用 Path 属性改变列表框的“当前目录”

图 8-9　FileListBox 控件

8.6.3　FileListBox 控件

FileListBox(文件列表框)控件用来显示指定目录中的部分或全部文件,如图 8-9 所示。下面介绍 FileListBox 控件的主要属性和事件。

1.Path 属性

Path 属性指定在运行时文件列表框列中要显示的文件所在的路径,其默认路径为系统当前工作路径。如可利用以下语句,在文件列表框中显示“c:\mydata”(假设 mydata 目录已存在)的所有文件(设文件列表框名为 File1):

```
File1.Path="c:\mydata"
```

重新设置 Path 属性会引发"PathChange"事件。

2.Pattern 属性

Pattern 属性用来确定程序运行时在文件列表框中要显示的文件类型。

Pattern 属性支持文件通配符“*”和“?”的使用。在默认情况该属性的值为“*.*”,即显示所有类型的文件。例如:

```
File1.Pattern="*.txt"          '在文件列表框中显示所有扩展名为".txt"的文件
File1.Pattern="???.txt"        '显示所有文件名长度为 3,扩展名为".txt"的文件
```

若要在文件列表框中显示多种类型的文件,还可以使用“;”将多种文件类型进行分割。例如:

```
File1.Pattern="*.txt;*.bas"    '显示所有扩展名为".txt"或".bas"的文件
```

重新设置 Pattern 属性会引发“PatternChange”事件。

3.FileName 属性

FileName 属性主要用于返回在文件列表框中被选定文件的文件名。

例如:

```
FName$ = File1.FileName
```

此语句执行后,变量 FName 中即为在文件列表框中被选定文件的文件名。须特别说明:用 FileName 属性来获取被选定文件的文件名时,并不包括该文件所在的驱动器和路径名。要从文件列表框中获得被选定文件的全路径的文件名,可用下面的程序代码实现:

```
If Right(File1.Path,1) ="\" Then
    FName$ = File1.Path & File1.FileName
```

```
Else
    FName $ = File1. Path & "\" & File1. FileName
End If
```

在程序运行过程中,还可以通过给 FileName 属性赋值的方式来设置在文件列表框中要显示文件所在路径和文件类型,此时功能类似 Path 属性和 Patter 属性。举例说明如下:

(1) File1. FileName = "stuinfo. dat"

该语句执行后,在文件列表框中只显示由 Path 指定路径下的文件"stuinfo. dat",该文件列表项被选定并突出显示。此时要求在文件"stuinfo. dat"必须存在,否则会出错。该语句执行后,FileName 属性的值为"stuinfo. dat",而且 Pattern 属性的值亦被修改为"stuinfo. dat"。

(2) File1. FileName = " * . txt; * . bas"

该语句功能类似于设置 Pattern 属性。执行该语句后,在文件列表框中显示由 Path 指定路径下的所有扩展名为". txt"和". bas"的文件。语句执行后,当 MultiSelect 属性设置为 0 时,FileName 属性的值为长度为 0 的字符串(列表框中未选中任何列表项),同时 Pattern 属性的值亦修改" * . txt; * . bas"。

(3) File1. FileName = "c: \ mydata \ * . txt; * . bas"

该语句功能相当于同时设置 Path 属性为"c: \ mydata", Pattern 属性为" * . txt; * . bas"。且语句执行,当 MultiSelect 属性设置为 0 时,FileName 属性的值是长度为 0 的字符串。

在程序运行期间还可以通过单击文件列表项来修改 FileName 属性的值,其值即为选定的文件名。

FileName 属性值的改变可以会产生 PathChange(若路径改变)、PatternChange(若模式改变)、DblClick(若指定存在的文件)事件。

4. 文件类型设置属性

FileListBox 控件还提供了 Archive、Normal、Hidden、System 及 ReadOnly 属性来指定要显示的文件类型属性。在缺省情况下 Archive、Normal 和 ReadOnly 属性为 True, Hidden 和 System 属性为 False。

例如,若在列表框中只显示只读文件,则直接将 ReadOnly 属性设置为 True,其他属性设为 False 即可。

5. List、ListCount、ListIndex、MultiSelect 属性

FileListBox 控件的 List、ListCount、ListIndex 和 MultiSelec 属性含义及使用方法与 ListBox 控件完全相同。在程序中对文件列表框中显示的文件进行操作时,就可能用到这些属性。

例如,以下程序段是将文件列表框(File1)中的所有文件名显示在窗体上。

```
For i = 0 To File1. ListCount - 1
    Print File1. List(i)
Next i
```

6. PathChange 事件

当 Patern 属性的值改变时,触发此事件。可使用 PathChange 事件过程来响应 FileListBox 控件中路径的改变。

7. PatternChange 事件

当 Pattern 属性的值改变时，触发此事件。

8. Click 事件

单击 FileListBox 控件中文件列表项时，触发此事件。

例如，Click 事件代码如下：

```
Private Sub File1_Click()
    MsgBox File1.FileName
End Sub
```

则当在文件列表框中单击选中某文件时，用对话框将该文件名输出。

9. DblClick 事件

双击 FileListBox 控件中文件列表项时，触发此事件。

例如，DblClick 事件代码如下：

```
Private Sub File1_DblClick()
  Dim Fname As String
  If  Right(file1.path,1)="\" Then
      Fname=file1.path & file1.filename
  Else
      Fname=file1.path & "\" & file1.filename
  End If
  RetVal=Shell(Fname, 1)      '执行程序
End Sub
```

则当双击文件列表框中的某个可执行文件时，执行该文件。

8.6.4　文件系统控件的组合

在实际应用中，通常将文件系统的驱动器列表框、目录列表框及文件列表框这三个控件组合在一起使用，使它们能够同步显示文件目录信息。而要使这三个控件关联起来同步进行显示，可以通过编写驱动器列表框和目录列表框控件的 Change 事件代码来实现。

例如，设驱动器列表框控件名为 Drive1、目录列表框控件名为 Dir1、文件列表框控件名为 File1，则可用如下代码实现这三个控件的同步显示：

(1) 驱动器列表框的 Change 事件代码

```
Private Sub Drive1_Change()
  Dir1.Path=Drive1.Drive      '更新目录列表框中的 Path 属性
End Sub
```

当用户更新驱动器列表框中显示的驱动器时(即其 Drive 属性发生改变时)，就触发该 Change 事件，利用 Drive 属性的值修改目录列表框的 Path 属性，使得目录列表框与驱动器列表框同步显示。

(2) 目录列表框的 Change 事件代码

```
Private Sub Dir1_Change()
  File1.Path=Dir1.Path        '更新文件列表框中的 Path 属性
End Sub
```

而当目录列表框的 Path 属性改变时，就触发目录列表框的 Change 事件，利用 Path 属性的值修改文件列表框的 Path 属性，使得文件列表框与目录列表框以及驱动器列表框同步显示。

8.6.5 应用实例

【例 8-4】 利用文件系统控件制作一个图片浏览器。

该图片浏览器，完成的主要功能为：

通过驱动器列表框、目录列表框及文件列表框来搜索指定格式的图片文件，并通过 Image 控件(详见第 9 章)浏览该图片，以及当双击图片时，调用“画图”程序可对其进行编辑。在运行中还设置了一个 ComboBox 控件来选择要搜索的图片文件的类型，设置了一个 TextBox 控件来显示当前选择的路径。

“图片浏览器”的运行界面及各控件的布局如图 8-10 所示。实例中各控件的主要属性及设置见表 8-9。

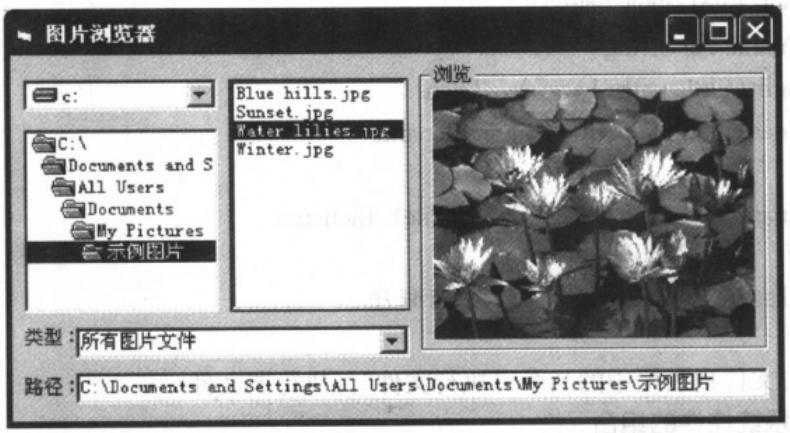


图 8-10 “图片浏览器”运行界面

表 8-9 “图片浏览器”的各控件及主要属性设置

对象	属性	属性值	说明
Form1	Caption	图片浏览器	Form 控件
Drive1			DriveListBox 控件，属性值为缺省值
Dir1			DirListBox 控件，属性值为缺省值
File1			FileListBox 控件，属性值为缺省值
Frame1	Caption	浏览	Frame 控件
Image1	Stretch	True	Image 控件，用来显示图片
Label1	Caption	类型：	Label 控件
Label2	Caption	路径：	Label 控件
Combo1	Text	(空)	ComboBox 控件，用来选择文件类型
Text1	Enabled	False	TextBox 控件，用来显示路径

本示例编写代码如下：

(1) 通用过程段内定义的变量。

```
Dim PaintName As String      '用来存储“画图”程序的文件名及路径
```

(2) 窗体的 Load 事件。

```
Private Sub Form_Load()
  '为组合框添加各个项目
  Combo1.AddItem "所有图片文件":Combo1.AddItem "*.bmp"
  Combo1.AddItem "*.jpg":Combo1.AddItem "*.jpeg"
  Combo1.AddItem "*.gif":Combo1.AddItem "*.ico"
  Combo1.ListIndex=0                    '设置运行时的组合框显示的初始项
  Text1.Text=Dir1.Path                  '设置运行时的文本框显示的初始路径
  File1.Pattern="*.bmp;*.jpg;*.jpeg;*.gif;*.ico"
                                        '设置初始显示时文件过滤模式
  PaintName="C:\Windows\System32\mspaint.exe"
                                        '指定画图程序文件名及所在路径
End Sub
```

(3) “驱动器列表框”的 Change 事件。

```
Private Sub Drive1_Change()
  On Error Resume Next
  Dir1.Path=Drive1.Drive                '使目录列表框与驱动器列表框同步
  If Err.Number > 0 Then                '当有错误发生时(如软驱中无磁盘)
    MsgBox "驱动器未准备好!", vbCritical
  End If
End Sub
```

(4) “目录列表框”的 Change 事件。

```
Private Sub Dir1_Change()
  File1.Path=Dir1.Path                  '使文件列表框与目录列表框同步
  Text1.Text=Dir1.Path                  '在文本框中显示当前选择的路径
End Sub
```

(5) “文件类型组合框”的 Click 事件,实现文件过滤功能。

```
Private Sub Combo1_Click()
  If  Combo1.ListIndex=0  Then          '当组合框中选择项为“全部图片文件”时
      File1.Pattern="*.bmp;*.jpg;*.jpeg;*.gif;*.ico"
                        '设置 File1 的 Patten 属性为所有图片文件格式
  Else
      File1.Pattern=Combo1.Text
                        '设置 File1 的 Patten 属性 Comgo1 指定图片文件格式
  End If
End Sub
```

(6) “文件列表框”的 Click 事件,实现图片浏览功能。

```
Private Sub File1_Click()
  Dim FName As String
```

```
    If  Right(File1.Path,1)="\" Then
        FName=File1.Path & File1.Filename
    Else
        FName=File1.Path & "\" & File1.Filename
    End If
    On Error Resume Next
    Image1.Picture=LoadPicture(FName)      '为 Image 控件加载图片
    If Err.Number>0 Then      '当有错误发生时(如文件为非图片格式文件)
        MsgBox "非图片文件!", vbCritical
    End If
End Sub
```

(7)“文件列表框”的 DblClick 事件,实现用“画图”程序编辑图片功能。

```
Private Sub Image1_DblClick()
    Dim FName As String, PathName As String, ProcID As Integer
    If Right(File1.Path, 1)="\" Then
        FName=File1.Path & File1.FileName
    Else
        FName=File1.Path & "\" & File1.FileName
    End If
    PathName=PaintName & " " & FName
    ProcID=Shell(PathName, vbNormalFocus)      '调用“画图”程序编辑图片
End Sub
```

习题八

1.简答题

(1) 什么是文件? 文件有哪几种类型? 它们的区别是什么?

(2) 文件操作有哪几步骤? 各步骤有何作用?

(3) 如何将文件系统控件组合起来使用?

2.填空题

(1) Visual Basic 提供的对数据文件的 3 种访问方式为随机访问方式、(　　)和二进制访问方式。

【出处】2003 年 4 月全国计算机等级考试二级 Visual Basic

(2) 在文件列表框(名为 File1)中如果要显示所有扩展名为“.TXT”和“.BAS”的文件,需要设置其 Pattern 属性,正确的设置方式为:File1. Pattern=(　　)。

(3) 在名称为 Form1 的窗体上画一个文本框,其名称为 Text1,在属性窗口中把文本框的 MultiLine 属性设置为 True,然后编写如下事件过程:

```
Private Sub Form_Click()
    Open "d:\test\smtext1.txt" For Input As #1
    Do While Not (      )
        Line Input #1, aspect $
```

```
        whole $ = whole $ + aspect $ + Chr(13) + Chr(10)
    Loop
    Text1.Text = whole $
    Close #1
    Open "d:\test\smtext2.txt" For Output As #1
    Print #1, (          )
    Close #1
End Sub
```

上述程序的功能是,把磁盘文件 smtext1.txt 的内容读到内存并在文本框中显示出来,然后把该文本框中的内容存入磁盘文件 smtext2.txt。请填空。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

3. 选择题

(1) 以下关于文件的叙述中,错误的是(　　)。

A. 顺序文件中的记录一个接一个地顺序存放

B. 随机文件中记录的长度是随机的

C. 执行打开文件的命令后,自动生成一个文件指针

D. LOF 函数返回给文件分配的字节数

【出处】2004 年 9 月全国计算机等级考试二级 Visual Basic

(2) 在窗体上画一个名称为 File1 的文件列表框,并编写如下程序:

```
Private Sub File1_DblClick()
    x = Shell(File1.FileName,1)
End Sub
```

以下关于该程序的叙述中,错误的是(　　)。

A. x 没有实际作用,因此可以将该语句写为:Call Shell(File1,FileName,1)

B. 双击文件列表框中的文件,将触发该事件过程

C. 要执行的文件的名字通过 File1.FileName 指定

D. File1 中显示的是当前驱动器、当前目录下的文件

【出处】2004 年 9 月全国计算机等级考试二级 Visual Basic

(3) 文件目录列表框的 Path 属性的作用是(　　)。

A. 显示当前驱动器或指定驱动器上的某目录下的文件名

B. 显示当前驱动器或指定驱动器上的目录结构

C. 显示根目录下的文件名

D. 显示指定路径下的文件

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(4) 以下叙述中正确的是(　　)。

A. 一个记录中所包含的各个元素的数据类型必须相同

B. 随机文件中每个记录的长度是固定的

C. Open 命令的作用是打开一个已经存在的文件

D. 使用 Input # 语句可以从随机文件中读取数据

【出处】2002 年 9 月全国计算机等级考试二级 Visual Basic

(5) 以下叙述中错误的是(　　)。

A. 用 Shell 函数可以调用能够在 Windows 下运行的应用程序

B. 用 Shell 函数可以调用可执行文件,也可以调用 VisualBasic 的内部函数

C. 调用 Shell 函数的格式应为:<变量名>=Shell(…)

D. 用 Shell 函数不能执行 DOS 命令

【出处】2003 年 4 月全国计算机等级考试二级 Visual Basic

(6) 执行语句 Open "Tel.dat" For Random As #1 Len = 50 后,对文件 Tel.dat 中的数据能够执行的操作是(　　)。

A. 只能写,不能读　　B. 只能读,不能写

C. 既可以读,也可以写　　D. 不能读,不能写

【出处】2003 年 9 月全国计算机等级考试二级 Visual Basic

(7) 以下能判断是否到达文件尾的函数是(　　)。

A. BOF　　B. LOC　　C. LOF　　D. EOF

【出处】2003 年 9 月全国计算机等级考试二级 Visual Basic

(8) 在窗体上画一个名称为 Drive1 的驱动器列表框,一个名称为 Dir1 的目录列表框。当改变当前驱动器时,目录列表框应该与之同步改变。设置两个控件同步的命令放在一个事件过程中,这个事件过程是(　　)。

A. Drive1_Change　　B. Drive1_Click　　C. Dir1_Click　　D. Dir1_Change

【出处】2004 年 4 月全国计算机等级考试二级 Visual Basic

(9) 窗体上有两个名称分别为 Text1、Text2 的文本框,一个名称为 Command1 的命令按钮。运行后的窗体外观如图 8-11。

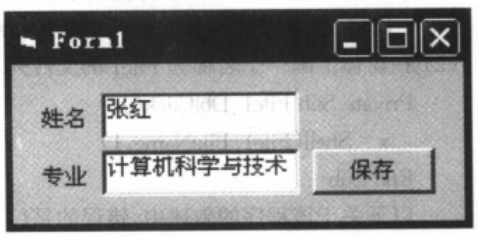


图 8-11　窗体外观

设有如下的类型声明

```
Type Person
  name As String * 8
  major As String * 20
End Type
```

当单击"保存"按钮时,将两个文本框中的内容写入一个随机文件 Test29.dat 中。设文本框中的数据已正确地赋值给 Person 类型的变量 p。则能够正确地把数据写入文件的程序段是(　　)。

A.
```
Open "c:\Test29.dat" For Random As #1
Put #1,1,p
Close #1
```

B.
```
Open "c:\Test29.dat" For Random As #1
Get # 1,1,p
Close #1
```

C.
```
Open "c:\Test29.dat" For Random As #1 Len=Len(p)
Put #1,1,p
Close #1
```

D.
```
Open "c:\Test29.dat "For Random As #1=Len(p)
Get #1,1,p
Close #1
```

【出处】2003 年 4 月全国计算机等级考试二级 Visual Basic

(10) 在窗体上画一个名称为 Command1 的命令按钮和一个名称为 Text1 的文本框,在文本框中输入以下字符串:

Microsoft Visual Basic Programming

然后编写如下事件过程：

```
Private Sub Command1_Click()
  Open "d:\temp\outf.txt" For Output As #1
  For i=1 To Len(Text1.Text)
    c=Mid(Text1.Text, i, 1)
    If c>="A" And c<="Z" Then
    Print #1, LCase(c)
    End If
  Next i
  Close
End Sub
```

程序运行后，单击命令按钮，文件 outf.txt 中的内容是(　　)。

A. MVBP　　B. mvbp

C. M　　D. m

V　　v

B　　b

P　　p

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

4. 编程题

(1) 某班级有 30 名同学，编程将他们的学号、姓名及 Visnal Basic 课程考试成绩输出到一个顺序文件中，文件名为“VBScore.dat”。

(2) 从第 1 题所建立的文件“VBScore.dat”中读入全部同学的记录信息。新建一个新的顺序文件“VBScorePass.dat”，将所有及格同学的记录信息输出到该文件中。

(3) 利用随机文件编写一个学生成绩管理程序，学生信息包括学号、姓名及 Visual Basic 课程考试成绩。要求能够实现学生成绩的录入、修改、删除等功能。

第 9 章　绘图应用程序设计

图形界面是 Windows 操作系统的一个显著特点，可视化图形操作为人机交互提供了友好、方便和快捷的交互环境，深受广大用户的欢迎。作为一种广泛使用的可视化程序开发工具，Visual Basic 除了提供窗体、按钮、文本框等可视化控件外，还为程序设计者提供了强大的图形处理功能。Visaul Basic 的图形处理功能主要体现在三个方面：一是利用窗体、图片框、图像框等图形控件来显示图形文件；二是利用 Line 和 Shape 两个基本控件来绘制一般的几何图形；三是采用绘图的方法及命令直接绘制图形。

9.1　图形操作基础

在绘制图形时，要达到成功的图形效果，就必需了解屏幕坐标系统、图形颜色及绘图相关属性等基本知识。

9.1.1　图形坐标系统

坐标系统用来描述屏幕、窗体、图片框等绘图区上的具体位置。一个坐标系统包括原点位置、坐标的度量单位、坐标轴及其方向这三个要素。在 Visual Basic 中，用户可以利用其提供的几种标准坐标系统，还可以自定义坐标系统。

1.标准坐标系统

在标准坐标系统中，每个对象左上角为坐标原点(0,0)，水平方向为 X 坐标轴且向右为正方向，垂直方向为 Y 坐标轴且向下为正方向。图 9－1 给出了 Form 对象和 PictureBox 对象的标准坐标系统。需要说明，每个对象使用的是它所在容器的坐标系统，如图 9－1 中 PictureBox 对象的位置(Left 和 Top 属性值)为 Form 对象坐标系统的坐标值，而 Text 对象的位置(Left 和 Top 属性的值)则采用 PictureBox 对象的坐标系统，而不是 Form 对象的坐标系统。

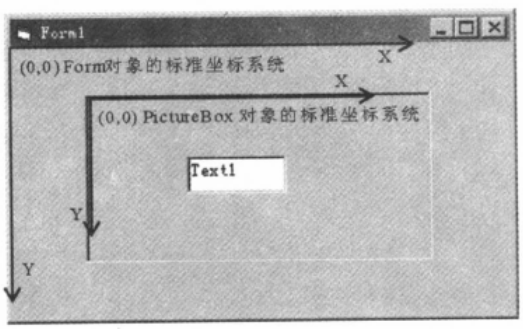


图 9－1　对象的标准坐标系统

在缺省时，Visual Basic 采用的默认坐标系统的度量单位（坐标刻度）为 Twips（缇）。除了 Twips 外，Visual Basic 还提供了 6 种标准规格的度量单位，用户可以根据实际需要使用 ScaleMode 属性来修改坐标系统的度量单位。表 9－1 给出 ScaleMode 属性的值及坐标系统采用的度量单位。

表 9－1　ScaleMode 属性的值及坐标系统采用的度量单位

属性值	坐标系统度量单位	说　明
0		用户自定义坐标系统。当采用自定义坐标系统时，该属性自动设置为 0
1	Twips	缇，为默认刻度单位，1 英寸＝1440 Twips
2	Point	磅，1 英寸＝72 磅
3	Pixel	像素
4	Char	字符，字符宽度＝120 twips，字符高度＝240 twips
5	Inch	英寸
6	Millimeter	毫米
7	Centimeter	厘米

例如：

```
Form1.ScaleMode＝3          '设置窗体坐标系的刻度单位为像素
```

2. 自定义坐标系统

在实际应用中，经常需要用户自行定义对象的坐标系统。在 Visual Basic 中，用户自定义坐标系统的方法有两种，一是使用对象的相关属性来设置坐标系统，二是采用 Scale 方法来设置坐标系统。

（1）使用对象属性自定义坐标系统。与自定义坐标系统相关的对象属性有 ScaleLeft、ScaleTop、ScaleWidth 及 ScaleHeight 四个属性。

对象的 ScaleLeft 和 ScaleTop 属性分别指定对象绘图区左上角 X 轴（水平方向）和 Y 轴（垂直方向）的坐标值，改变原点的位置。ScaleHeight 和 ScaleWidth 属性用来指定对象绘图区的高度与宽度。在这里“绘图区”是指对象去除边界（包括标题行）的内部区域。另外，若改变对象的这 4 个属性中的任一个，则其 ScaleMode 属性自动置为 0。

例如，用如下语句定义窗体的坐标系统（设窗体名为 Form1）：

```
Form1.ScaleWidth＝5         '将窗体的可用宽度设为 5（当前窗体宽分为 5 份）
Form1.ScaleHeight＝4        '将窗体的可用高度设为 4（当前窗体高度分为 4 份）
Form1.ScaleLeft＝－2        '窗体的左上角的横坐标设为－2
Form1.ScaleTop＝－1         '窗体的左上角的纵坐标设为－1
```

执行以上语句后，窗体中的坐标系统如图 9－2(a)所示。若再执行如下两个语句：

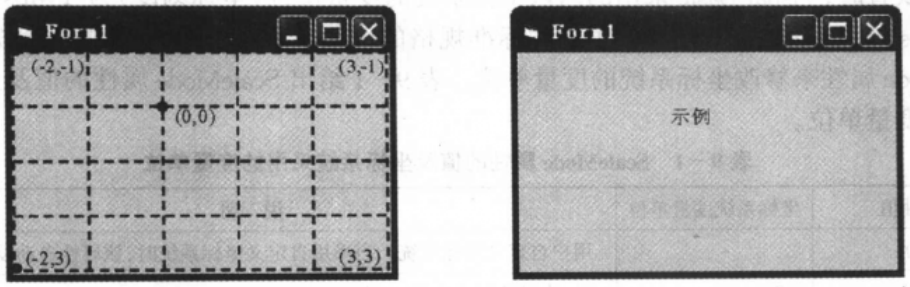
```
Cls                         '清除窗体，输出位置设在(0,0)点
Print "示例"                 '在(0,0)点输出示例
```

执行结果如图 9－1(b)所示。

（2）采用 Scale 方法定义坐标系统。Scale 方法的使用格式为：

[对象.] Scale [(x1,y1)－(x2,y2)]

说明：



(a) (b)

图 9-2 用户自定义窗体坐标系示意图

① (x1,y1)和(x2,y2)分别为左上角和右下角的坐标。

② 格式中的"对象."可以省略,若省略默认对象为当前窗体。

③ 格式中的"(x1,y1)-(x2,y2)"也可以省略,此时恢复为默认坐标系统。

例如,采用如下语句定义窗体坐标系统:

Form1.Scale (-2,-1)-(3,3)

则此语句执行后,仍可得到如图 9-2(a)所示的坐标系统。

9.1.2 绘图颜色

进行绘图操作时自然就离不开颜色。在 Visual Basic 中关于对象颜色属性(如 ForeColor 与 BackColor 属性)的设置有两种方式:一种是在设计阶段时使用"调色板"来选择颜色或直接在属性窗口中单击相应的属性进行设置;另一种是在程序运行中通过对颜色属性指定颜色值(每一种颜色都有一个颜色值与之对应)的方式进行颜色设置。在程序运行中,可以使用预先定义好的颜色常量来指定一种颜色,如 VbRed(红色)、VbGreen(绿色)、VbBlue(蓝色)、VbYellow(黄色)、VbWhite(白色)、VbBlack(黑色)等;也可以通过 Visual Basic 提供的 RGB 或 QBColor 函数来生成一种所需要的颜色。

下面介绍 RGB 函数和 QBColor 函数。

1.RGB 函数

RGB 函数通过三原色(红、绿、蓝)的值设置一种混合颜色,其格式为:

颜色值=RGB(红色值,绿色值,蓝色值)

其中,这三个原色值均为整数,取值范围为 0~255,表示混合色中每一种颜色的强度。某个参数的值越大,在混合中该原色越强。例如,RGB(0,0,0)黑色,RGB(255,255,255)表示白色,RGB(0,0,255)表示蓝色,RGB(255,255,0)表示黄色。RGB 函数返回颜色的总数为 $256^3=16M$,即真彩色模式下的颜色总数,当然这与系统的设置有关。

2.QBColor 函数

QBColor 函数也可以用于产生一种颜色,其使用格式为:

颜色值=QBColor (n)

其中,参数 n 的取值为 0~15,分别代表 16 种基本颜色。QBColor 函数与颜色之间的对应关系见表 9-2。

表 9-2　QBColor 函数与颜色的关系

函数	颜色	函数	颜色	函数	颜色	函数	颜色
QBColor(0)	黑色	QBColor(4)	红色	QBColor(8)	灰色	QBColor(12)	亮红色
QBColor(1)	蓝色	QBColor(5)	洋红色	QBColor(9)	亮蓝色	QBColor(13)	亮洋红色
QBColor(2)	绿色	QBColor(6)	黄色	QBColor(10)	亮绿色	QBColor(14)	亮黄色
QBColor(3)	青色	QBColor(7)	白色	QBColor(11)	亮青色	QBColor(15)	亮白色

9.1.3　绘图属性

要让绘制在 Form(窗体)或图片框(PictureBox)控件上的图形能长久保持,或绘出效果良好的图形,就必须掌握它们的几个关键属性。

1. CurrentX 属性和 CurrentY 属性

这两个属性用来获取或设置绘图区上当前的画笔位置坐标。其中,CurrentX 属性表示画笔的水平坐标,CurrentY 属性表示画笔的垂直坐标,这两个属性的默认值为 0。

例如,如下语句可以实现在窗体指定位置(1000,1000)用 Print 方法输出数据:

```
Form1.CurrentX=1000
Form1.CurrentY=1000
Form1.Print "示例"
```

2. AutoRedraw 属性

AutoRedraw 属性的默认值为 False,此时在窗体或图片框上显示的图形或文字都不保存在内存中,因此一旦图形被其他对象暂时挡住,或当窗体大小发生变化时,这些图形将会丢失。此时要重现原来的图形或文字,则必须重画,需激活 Paint 事件。

当 AutoRedraw 属性值为 True 时,则窗体或图片框上显示的图形或文字都保存在内存中,要重现窗体或图片框的内容,只需要将内存中保存的图形或文字图像调出即可。因此当 AutoRedraw 属性值为 True 时不会造成图形丢失的问题。

3. DrawWidth 属性

DrawWidth 属性用来设置绘图线的宽度,其单位为像素,取值范围是 1～32767,缺省值为 1。在程序运行过程中,可以通过改变 DrawWidth 属性的值来设置图线的宽度。

例如,在窗体(Form1)的 Paint 事件代码如下:

```
Private Sub Form_Paint()
  For i=1 To 4
    Form1.DrawWidth=i   '设置图线宽度
    Line (200, 200 * i)-(2900, 200 * i) '画线
    Circle (500+700 * (i-1),1300),300   '画圆
  Next i
End Sub
```

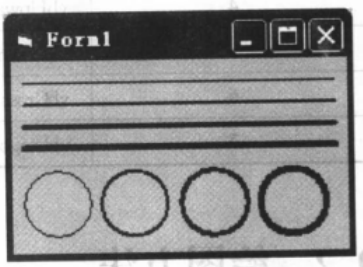


图 9-3　DrawWidth 属性的应用示例

图 9-3 即给出程序运行后的结果。

4. DrawStyle 属性

DrawStyle 属性用于指定绘图时图线的类型(实线或虚线),它可以取 0～6 不同值(缺省值为 0,实线)。

例如，下面的程序将演示 DrawStyle 属性设置值的效果，程序运行结果如图 9－4 所示。

```
Private Sub Form_Paint()
    Dim a As Integer, y As Integer, st As String
    For a=0 To 6
        DrawStyle=a
        y=300 * a+350
        st="DrawStyle=" & a
        CurrentX=200
        CurrentY=y－TextHeight(st)/2
        Print st
        Line(200+TextWidth(st)+200, y)－(4000, y)
    Next a
End Sub
```

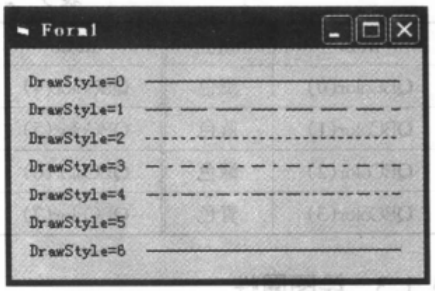


图 9－4 DrawStyle 属性的设置效果

注意，只有当 DrawWidth 属性值设置为 1 时，DrawStyle 属性设置值才全部起作用。

5. FillColor 属性和 FillStyle 属性

利用 FillColor 属性和 FillStyle 属性可以对已绘制好的封闭图形设置填充方案。

FillColor 属性用来指定填充颜色，FillStyle 属性用来指定填充样式(图案)。FillStyle 属性可以取 0～7 的值，默认值为 1(透明，不填充)，表 9－3 给出了 FillStyle 属性的值及其表示的填充样式。

表 9－3 FillStyle 属性的值及其表示的填充样式

属性值	常 量	说 明	示 例
0	vbFSSolid	实心	
1	vbFSTransparent	透明(默认值)	
2	vbHorizontalLine	水平线	
3	vbVerticalLine	垂直线	
4	vbUpwardDiagonal	左上对角线	
5	vbDownwardDiagonal	右下对角线	
6	vbCross	交叉线	
7	vbDiagonalCross	对角交叉线	

9.2 绘图方法

Visual Basic 提供了 4 种基本绘图方法，分别为：PSet、Line、Circle 和 Point 方法。

9.2.1 PSet 方法

PSet 方法可以在对象的指定位置画点，其使用格式为：

[对象名.] PSet [Step] (x,y) [,颜色]

说明：

(1) “对象名”为要绘制点的对象名，可以是窗体和图片框，若省略则为当前窗体。

(2) “[step](x,y)”为位置参数，用来指定画点位置。Step 为关键字，如果省略，则“(x,y)”指的是绝对坐标，即对象坐标系统上的位置；否则“(x,y)”指的是相对于当前画笔位置(CurrentX,CurrentY)点的位移量，即画点位置坐标为(CurrentX+x,CurrentY+y)。

(3) “颜色”用来指定要绘制点的颜色，它可以是颜色常量或颜色函数。若省略，则默认颜色为对象的前景色(ForeColor)。

(4) 利用 PSet 方法所画点的大小取决于当前容器的 DrawWidth 属性值。

9.2.2 Line 方法

Line 方法用来画直线和矩形，其格式为：

[对象名.] Line [[Step] (x1,y1)] - [Step] (x2,y2) [,颜色] [,B[F]]

说明：

(1) 格式中“[Step] (x1,y1)”与“[Step] (x2,y2)”用来指定线段或矩形(带有参数 B 时)的起点和终点坐标。关键词“Step”可以省略，此时坐标为对象坐标系统上的绝对坐标。若起点(x1,y1)前有 Step，则(x1,y1)表示起点坐标相对于当前画笔位置的位移量。若在终点(x2,y2)前有 Step，则(x2,y2)表示相对起点坐标的偏移量。另外起点坐标“[Step] (x1,y1)”也可以省略，此时起点为当前画笔位置，即(CurrentX,CurrentY)。

(2) 当省略参数 B 时，则从起点到终点画一直线段。若带有参数 B，则画一矩形，此时(x1,y1)与 (x2,y2)表示所画矩形的对角坐标。当带有参数 B 时还可以带有参数 F，若省略参数 F，则所画矩形以对象当前指定(或默认)的填充方式(默认为透明)进行填充；若带有参数 F，则所画矩形以其边框颜色填充。

(3) 其他参数与画点 PSet 方法相同。

例如：

```
Line (200, 200) - (3500, 500)          '画直线
Line (200, 800) - (1700, 2500), , B    '画矩形，当前填充方式填充
Line (2000, 800) - (3500, 2500), , BF  '画矩形，并以边框颜色填充
```

9.2.3 Circle 方法

Circle 方法用来画圆、椭圆、圆弧、扇形等，其使用格式为：

[对象名.] Circle [Step] (x,y), 半径 [,颜色] [, 起始角,终止角] [,纵横比]

说明：

(1) “[Step] (x,y)”为指定圆点坐标，缺省 Step 时则为容器对象的绝对坐标，否则为相对于当前画笔位置的相对坐标。

(2) 格式中的“半径”用来指定所画圆(或圆弧、扇形)的半径长度，或为椭圆的长半轴长度。

(3) 可选参数“起始角,终止角”用来指定弧或扇形的起始角和终止角，其单位为弧度。当参数为正时则画弧，当参数为负时则从圆心到负端点连接一条线(可用于画扇形)。当该参数省略时，则默认为画圆(或椭圆，由参数“纵横比”指定)。

(4) 可选参数“纵横比”用于指定水平长度和垂直长度的比值，可用于画椭圆。省略时，则默认为 1，即画圆。

(5) 其他参数与画点 PSet 方法相同。

(6) Circle 方法执行后,当前画笔位置(CurrentX 和 CurrentY 属性)位于圆心位置。

例如,下面的程序演示了 Circle 方法画圆、圆弧、扇形及椭圆的方法。程序运行结果如图 9-5 所示。

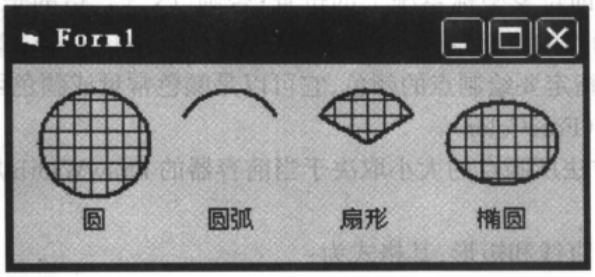


图 9-5 用 Circle 方法绘图

```
Private Sub Form_Paint()
  Const Pi=3.1415926
  Dim i As Integer, str(3) As String
  str(0)="圆":str(1)="圆弧":str(2)="扇形":str(3)="椭圆"
  DrawWidth=2:FillStyle=6          '设置线宽及图形填充方案
  Circle (600, 600), 400           '画圆
  Circle (1600, 600), 400, , 30*Pi/180, 150*Pi/180      '画圆弧
  Circle (2600, 600), 400, , -30*Pi/180, -150*Pi/180    '画扇形
  Circle (3600, 600), 400, , , , 0.8      '画椭圆
  For i=0 To 3
    CurrentX=600+i*1000-TextWidth(str(i))/2:CurrentY=1100
    Print str(i)
  Next i
End Sub
```

9.2.4 Point 方法

Point 方法的格式为:

[对象名.] Point (x,y)

Point 方法用于返回指定位置为(x,y)处的颜色值,若(x,y)位于对象之外,则返回值为 -1。

9.2.5 应用实例

【例 9-1】 用 PSet 和 Line 方法绘制正弦函数 sin(x)与余弦函数 cos(x)图形。程序运行界面如图 9-6 所示。

本例用到了 1 个窗体(From1)控件和 1 个命令按钮(Commannd1)控件。Command1 的 Click 事件代码编写如下:

```
Private Sub Command1_Click()        'Command1 的 Click 事件,实现绘图功能
  DimxAs Single, y As Single, alfa As Integer
  Const Pi=3.1415926
  DrawWidth=1                       '置图线宽度为 1,用于画坐标轴
```

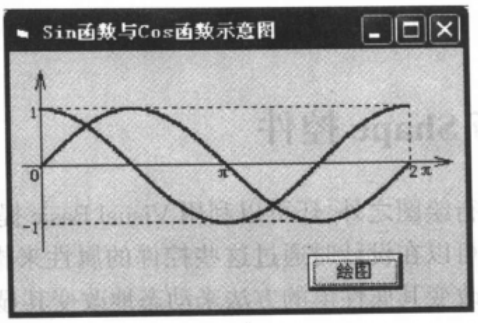


图 9-6　例 9-1 运行界面

```
Line (100, 1200)-(4500, 1200)    '画 X 坐标轴,以下两个语句画 X 轴箭头
Line (4500, 1200)-(4300, 1150)
Line (4500, 1200)-(4300, 1250)
Line (300, 200)-(300, 2100)      '画 Y 坐标轴,以下两个语句画 Y 轴箭头
Line (300, 200)-(250, 400)
Line (300, 200)-(350, 400)
DrawWidth=2                      '置图线宽度为 2,用于画正弦函数与余弦函数图像
For alfa=0 To 360                '利用循环画正弦函数与余弦函数上的点
  x=alfa * Pi/180 * 600+300
  y=-1 * 600 * Sin(alfa * Pi/180)+1200
  PSet (x, y), vbBlack           '以黑色画 Sin 函数上的点
  y=-1 * 600 * Cos(alfa * Pi/180)+1200
  PSet (x, y), vbRed             '以红色画 Cos 函数上的点
Next alfa
DrawWidth=1:DrawStyle=2          '重置线宽度为 1,线型为 2(虚线)
x=360 * Pi/180 * 600+300
Line (x, 600)-(x, 1200)
Line (300, 600)-(x, 600)
x=270 * Pi/180 * 600+300
Line (300, 1800)-(x, 1800)       '以下语句用于输出图形坐标刻度
CurrentX=200:CurrentY=550
Print "1"
CurrentX=200:CurrentY=1220
Print "0"
CurrentX=100:CurrentY=1750
Print "-1"
CurrentX=Pi * 600+180:CurrentY=1220
Print "π"
CurrentX=2 * Pi * 600+300:CurrentY=1220
```

```
    Print "2π"
End Sub
```

9.3 Line 控件与 Shape 控件

除了利用绘图方法进行绘图之外，还可以利用 Visual Basic 提供的 Line 和 Shape 图形控件来绘制简单图形。用户可以在设计时通过这些控件的属性来指定其显示位置及形式，也可以在程序运行期间通过改变其属性值的方法来动态地改变其显示位置及形式。

9.3.1 Line 控件

利用 Line(直线)控件，可以在图形容器(如窗体、图片框等)上画各种直线段。其主要属性如下：

1.坐标属性

Line 控件的 X1、Y1、X2、Y2 属性用来指定线段在容器对象上的起点(X1，Y1)和终点(X2，Y2)位置。

2.BorderStyle 属性

BorderStyle 属性用来设置直线的线型，该属性的取值范围为 0～6。表 9－4 给出 BorderStyle 属性值及其所表示的线型。

表 9－4 BorderStyle 属性值及其表示的线型。

属性值	说 明	示 例
0	透明线	
1	实线(默认值)	————————
2	长划线	— — — — — — — —
3	点线	----------
4	点划线	—·—·—·—·—·—
5	双点划线	—··—··—··—··—
6	内部实线	————————

3.BorderWidth 属性

BorderWidth 属性与容器的 DrawWidth 类似，用于指定线条的宽度。

4.BorderColor 属性

BorderColor 属性用来指定线段的颜色。

9.3.2 Shape 控件

Shape(形状)控件用来绘制常见的几何图形。可以利用 Shape 控件的 Shape 属性指定该控件要显示的图形形状，该属性的取值范围为 0～5。表 9－5 给出了 Shape 控件的 Shape 属性值及其表示的图形形状。

Shape 控件的 FillColor 和 FillStyle 属性用来指定图形的填充颜色和填充图案方式，这两个属性的取值范围及含义与容器(如窗体)的 FillColor 和 FillStyle 相同。

另外，还可用 BorderStyle、BorderWidth 及 BorderColor 属性设置图形的边框样式、宽度及颜色。这些属性的使用方式与 Line 控件相同。

表 9－5　Shape 控件的 Shape 属性值及其表示的图形形状

属性值	常　量	说　明	示　例
0	vbShapeRectangle	矩形(默认值)	
1	vbShapeSquare	正方形	
2	vbShapeOval	椭圆	
3	vbShapeCircle	圆	
4	vbShapeRoundedRectangle	圆角矩形	
5	vbShapeRoundedSquare	圆角正方形	

9.4　Image 控件和 PictureBox 控件

Image 控件和 PictureBox 控件是 Visual Basic 提供的两个用于显示图片的控件，而且 PictureBox 控件还可以作为容器控件，在其中添加其他控件或绘制图形、输出文字等等。在用于显示图片时，Image 控件比 PictureBox 控件显示速度快，因此在只需显示图片的情况下，应优先考虑 Image 控件。

9.4.1　Image 控件

Image 控件用于显示保存在图形文件中的图像，它支持的图形文件格式有位图文件(*.bmp 或 *.dib)、图标文件(*.ico)、图元文件(.wmf 或.emf)、JPEG 文件(.jpg 或.jpeg)以及 GIF 文件(.gif)等。

Image 控件除了具有在含义和使用方法上与其他控件相同的属性(如，Name 属性、Left 属性、Top 属性、Width 属性、Height 属性、Visible 属性、BorderStyle 属性及 Enabled 属性)外，还有两个比较重要的属性，Picture 属性和 Stretch 属性。

(1) Picture 属性。与窗体的 Picture 属性的使用方法相同，Image 控件的 Picture 属性用来指定在图像框中所显示的图像来源。在程序设计时，可以单击属性窗口的 Picture 属性，通过选择图片文件的方式来指定所显示的图片内容。在程序运行阶段，可以用内部函数 LoadPicture 给 Picture 属性赋值的方式将一个图形文件加载到图像框中，也可以利用其他 Image 控件(或 Form、PictureBox 控件)的 Picture 属性来指定要显示的图片内容。例如：

```
Image1.Picture = LoadPicture ("C:\pic1.jpg")
Image2.Picture = Image1.Picture      'Imgae2 显示的图片内容与 Image1 相同
```

在使用 LoadPicture 函数时，函数的参数应该包括要显示的图形文件的完整路径和文件名。当参数为空或空串("")时，则可以利用 LoadPicture 函数卸载图像框中的图片文件。例如：

```
Image1.Picture = LoadPicture()
Image2.Picture = LoadPicture("")
```

在默认情况下，Picture 属性不指定任何图像文件，图像框中不显示任何图片。

(2) Stretch 属性。当 Stretch 属性为 True 时，加载的图片能够自动调整尺寸以适应图

像框的大小，此时可能会造成图片失真。当 Stretch 属性为 False 时(默认值)，则图像框将自动改变大小以适应其中的图形。

9.4.2 PictureBox 控件

PictureBox 控件也可以用于显示图片，与 Image 控件不同的是 PictureBox 控件还可以作为容器控件，用来放置其他控件或在其上绘图、输出文本(利用 Print 方法)等。

PictureBox 控件的 Picture 属性及使用方法与 Image 控件相同。PictureBox 控件没有 Stretch 属性，但提供了与 Image 控件 Stretch 属性类似的 AutoSize 属性。

PictureBox 控件的 AutoSize 属性用于调整图像框的大小以适应图形尺寸。AutoSize 属性值为 False(默认)时，图像框保持原尺寸，若图片比图片框大时，超出部分被截去不显示。AutoSize 属性值为 True 时，图片框根据图片大小自动调整。

9.4.3 应用实例

【例 9-2】 利用图像框制作一个图片缩放器。程序运行界面及各控件的布局如图 9-7 所示。

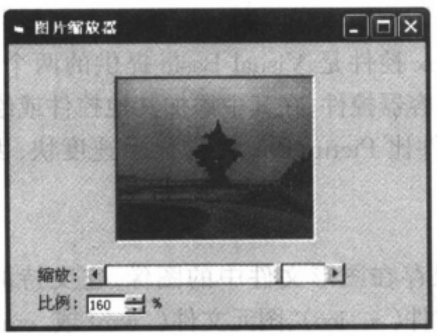


图 9-7 “图片缩放器”运行界面

实例中用到的控件及各控件的主要属性设置见表 9-6。

表 9-6 “图片缩放器”中的各控件及其主要属性设置

对 象	属 性	属性值	说 明
Form1	Caption	图片缩放器	Form 控件
Image1	Stretch	True	Image 控件
	BorderStyle	1	
HScroll1	Max	200	HScrollBar 控件
	Min	0	
	Value	100	
	LargeChange	10	
	SmallChange	2	
Text1	Text	(空)	Text 控件，用于显示当前缩放比例
Lable1	Caption	缩放：	
Lable2	Caption	比例：	
Lable3	Caption	%	

续表 9-6

对　象	属　性	属性值	说　明
UpDown1	Max	200	UpDown 控件，用于精确设置显示比例。此控件包含在 Microsoft Windows Common Control -2 6.0 部件中
	Min	0	
	Value	100	
	Increment	1	

该实例的程序代码如下：

```
Dim Picture_Height As Integer, Picture_Width As Integer    '用于存储图像框初始高度与宽度
Dim x0 As Integer, y0 As Integer    '用于存储图像框的中心位置坐标，以便按中心进行缩放
Private Sub Suofang(num As Single)    '实现缩放功能，比例为 num/100
  Image1.Width = Picture_Width * num/100    '缩放后的高度
  Image1.Height = Picture_Height * num/100    '缩放后的宽度
  Image1.Left = x0 - Image1.Width/2    '缩放后图像框的 Left 属性值
  Image1.Top = y0 - Image1.Height/2    '缩放后图像框的 Top 属性值
End Sub
Private Sub Form_Load()    '窗体的 Load 事件
  Picture_Height = Image1.Height    '求图像框初始高度
  Picture_Width = Image1.Width    '求图像框初始宽度
  x0 = Image1.Left + Image1.Width/2    '求图像框中心位置的 X 坐标
  y0 = Image1.Top + Image1.Height/2    '求图像框中心位置的 Y 坐标
End Sub
Private Sub HScroll1_Change()    'HScroll1 的 Change 事件
  Dim a As Single
  a = HScroll1.Value
  Text1.Text = a: UpDown1.Value = a    '使 Text1 和 Updown1 与 HScroll1 同步
  Suofang (a)    '按 a/100 的比例进行缩放
End Sub
Private Sub HScroll1_Scroll()    'HScroll1 的 Scroll 事件
  HScroll1_Change    '调用 Change 事件，实现缩放
End Sub
Private Sub UpDown1_DownClick()    '单击 UpDown1 的向下箭头时发生此事件
  Text1.Text = UpDown1.Value    '使 Text1 与 Updown1 同步
  HScroll1.Value = UpDown1.Value    '使 T HScroll1 与 Updown1 同步
End Sub
Private Sub UpDown1_UpClick()    '单击 UpDown1 的向上箭头时发生此事件
  Text1.Text = UpDown1.Value
```

```
    HScroll1.Value = UpDown1.Value
End Sub
```

习题九

1. 简答题

(1) 自定义坐标系统的方法有哪几种？如何使用？

(2) 绘图容器对象的 AutoRedraw 属性的作用是什么？

(3) 如何使用 Line 方法画直线与矩形？

(4) 怎样用 Circle 方法画圆、圆弧、扇形和椭圆？

(5) Image 控件与 PictureBox 有什么区别？

2. 填空题

(1) 坐标系统的 3 个要素为(　　)、(　　)和(　　)。

(2) 将 C 盘根目录下的图形文件 moon.jpg 装入图片框 Picture1 的语句是(　　)。

【出处】2004 年 4 月全国计算机等级考试二级 Visual Basic

(3) 在窗体上画一个文本框和一个图片框，然后编写如下两个事件过程：

```
Private Sub Form_Click()
    Text1.Text = "VB 程序设计"
End Sub
Private Sub Text1_Change()
    Picture1.Print "VBProgramming"
End Sub
```

程序运行后，单击窗体，在文本框中显示的内容是(　　)，而在图片框中显示的内容是(　　)。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(4) 执行以下语句后：

```
Form1.ScaleLeft = 200: Form1.ScaleTop = 300
```

窗体 Form1 的左上角坐标为(　　)。

(5) 若在当前窗体的坐标为(400, 500)位置处开始输出文字“示例”，则应使用的语句为：(　　):Print "示例"

(6) RGB 函数的 3 个参数分别用来表示在混合色中三基色(　　)、(　　)和(　　)的强度，其值的范围为(　　)。

(7) Point(x,y)方法的功能是用来获取(x,y)处的(　　)。

3. 选择题

(1) 假定在图片框 Picture1 中装入了一个图形，为了清除该图形(不删除图片框)，应采用的正确方法是(　　)。

A. 选择图片框，然后按 Del 键

B. 执行语句 Picture1.Picture = LoadPicture("")

C. 执行语句 Picture1.Picture = ""

D. 选择图片框，在属性窗口中选择 Picture 属性，然后按回车键

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(2) 以下关于图片框控件的说法中，错误的是(　　)。

A. 可以通过 Print 方法在图片框中输出文本

B. 清空图片框控件中图形的方法之一是加载一个空图形

C. 图片框控件可以作为容器使用

D. 用 Stretch 属性可以自动调整图片框中图形的大小

【出处】2004 年 4 月全国计算机等级考试二级 Visual Basic

(3) 图像框有一个属性,可以自动调整图形的大小,以适应图像框的尺寸,这个属性是(　　)。

A. Autosize　　B. Stretch　　C. AutoRedraw　　D. Appearance

【出处】2002 年 9 月全国计算机等级考试二级 Visual Basic

(4) QBColor(5)表示的的颜色为(　　)。

A. 洋红色　　B. 红色　　C. 蓝色　　D. 黄色

(5) 使用 Cls 语句清除窗体时,窗体当前的坐标位置为(　　)。

A. 窗体的左上角　　B. 窗体中心位置　　C. (0,0)　　D. 窗体的右下角

(6) 绘图容器控件的(　　)属性用来指定绘图的线型。

A. DrawWidth　　B. DrawStyle　　C. FillStyle　　D. FillColor

(7) 对象的边框类型由属性(　　)设置。

A. DrawStyle　　B. DrawWidth　　C. BorderStyle　　D. ScaleMode

4. 编程题

(1) 在窗体中绘制函数 $y=9/x$ 的曲线。其中 x 的取值区间为[-20,-1]和[1,20]。

提示:可通过自定义坐标系统的方法将窗体绘图区的中心位置定义为坐标原点(0,0),并定义适当的坐标刻度。

(2) 在窗口上画出奥运五环,并在五环上输出"北京 2008"字样。

第 10 章　数据库管理应用程序设计

计算机在信息处理方面，都会涉及到数据库技术的应用，Visual Basic 最引人注目的特点之一就是它访问数据库的强大功能。在 Visual Basic 中，通过数据库控件和数据访问对象可以实现对数据库的访问。特别是 Visual Basic 6.0 引入了功能强大的 ADO 作为存储数据的新标准，并提供了新的数据环境设计器，使数据访问更为灵活和简便。Visual Basic 可以访问下列数据库：

Jet 数据库(Visual Basic 自带的数据库)，即 Microsoft Access 数据库，是 Visual Basic 6.0 处理的默认数据库。

ISAM 数据库(外部数据库)，如 DBASE、FoxPro 和 TextFiles 等。

ODBC 数据库(凡是遵循 ODBC 标准的客户/服务器数据库)，如 Microsoft SQL Server 和 Oracle 等。

本章主要介绍利用 Visual Basic 访问数据库，实现数据库管理应用程序的基本方法。

10.1　数据库基础知识

10.1.1　数据库概念

数据库是指以一定的组织形式存放在计算机存储介质上相互关联的数据集合。例如，把一个学校的学生、教师和课程等数据有序地组织起来，存储在计算机磁盘上，可以构成一个数据库。此后用户可随时查询到该数据库的有关信息。

1.数据库的特点

数据库具有以下特点：

(1) 具有最小的冗余度，即数据尽可能功能不重复。

(2) 资源共享性，即以最优的方式服务于一个或多个应用程序。

(3) 数据独立性，即数据的存储尽可能独立于使用它的应用程序。

(4) 安全可靠。

(5) 保密性能好。

2.数据库的类型

按数据的组织方式不同，数据库可以分为网状数据库、层次数据库和关系型数据库三种类型。其中应用最为普遍的是关系型数据库，大至 Sybase、Oracle、SQL Server，小到 Access、FoxPro、DBase 等都是关系型数据库。

3.数据库管理系统

数据库管理系统(DBMS)是对数据库进行管理的系统软件，是用户与数据库之间的接口，如 Access、Visaul FoxPro、Oracle、DB2、Sybase、SQL Server 等。一个数据库管理系统提供

了用户对数据库进行操作的各种命令、工具及方法，包括数据库的建立和数据的显示、维护、查询、统计等方面。

10.1.2　关系数据库及其结构

关系数据库是以关系模型为基础的数据库，是根据表、记录和字段之间的关系进行组织和访问的一种数据库。关系型数据库通过若干个二维表来存取数据，并且通过关系将这些表联系在一起。

1.表(Table)

关系数据库包括若干个表，表是数据的集合，它是由行列方式组织的二维表格。表是一种数据对象，它可以具有很多属性，这些属性构成了表的结构。例如，表 10－1 所示的学生基本信息表“StuInfo”就包括了“学号”、“姓名”、“性别”、“出生日期”、“家庭地址”及“入学成绩”等属性。

表 10－1　学生基本信息表(StuInfo)

学　号	姓　名	性　别	出生日期	家庭地址	入学成绩
08060301	刘永利	男	1988.6.12	黑龙江	513
08060302	孙丽丽	女	1987.11.06	北京	546
08060303	张晓明	男	1988.09.23	辽宁	556
08060304	周大伟	男	1988.04.16	浙江	532
08060305	李婧	女	1987.03.24	吉林	498

2.字段(Field)

数据库表中的每一列称为字段，一个字段用来标记实体的一个属性。每个字段有一个字段名称。创建一个数据库表时，要设置每个字段的数据类型等属性。表 10－1 所示的学生基本信息表“StuInfo”即包括了“学号”、“姓名”、“性别”、“出生日期”、“家庭地址”及“入学成绩”6 个字段。其中，“姓名”字段可定义为字符型，“出生日期”可定义为日期型，“入学成绩”可以定义为整型。

3.记录(Record)

数据库表中的每一行称为一个记录。一个记录用来存储一个实体的相关信息。如表 10－1 所示的学生基本信息表“StuInfo”就有 5 个记录，分别用来表示 5 个同学的信息。每个同学的信息用 6 个字段来表示。

4.关键字(Key)与主关键字(Primary Key)

在数据库表中能够用来标识(识别)一个记录的某个字段或多个字段的组合称为关键字。若此关键字能够惟一地标识一个记录，则称之为主关键字。即主关键字必须有一个惟一的值，且不能为空值。例如，在表 10－1 所示的“StuInfo”表中，可以利用“学号”和“姓名”字段作为关键字。对于每个同学其学号是惟一的，因此“学号”字段可以作为主关键字，而“姓名”字段(可以有同名的现象)则不能作为主关键字。

5.索引

大多数数据库都使用索引，其目的在于提高检索数据库记录的效率。索引是根据数据库表中关键字所建立的，它以特定的顺序将关键字全部的值及其相应记录的地址存储在一个索引文件上。实际上，索引就是关键字的值到记录位置的一张转换表。在一个数据库表

中可以建立多个索引，但只能有一个主索引，且主索引的字段值不允许重复，是惟一的。

10.1.3 SQL 语言

SQL(Structure Query Language)语言，称为结构化查询语言，它已经成为数据库语言的通用标准。关系数据库管理系统都支持 SQL 语言，利用 SQL 语言可以方便快捷地实现对数据库的操作(包括创建、查询、插入、删除、修改等)，其常用的数据库操作语句有 SELECT、CREATE、INSERT、DELETE、UPDATE 等。

1. SELECT 语句

SELECT 语句用于从指定的表中查询满足给定条件的记录。其基本语法格式为：

SELECT 字段名表 FROM 表名 [WHERE 条件]

说明：

(1) “FROM 表名”为 FROM 子句，用来指定要查询的数据库表。

(2) “字段名表”，用来指定从给定的数据库表中要获取哪些字段。当要选定数据表的所有字段时，可用“ * ”表示。

(3) “WHERE 条件”为 WHERE 子句，用来指定查询条件，其中“条件”可以是一个关系表达式。

例如：

(1) SELECT * FROM StuInfo WHERE 性别 = '男'

在学生信息表“StuInfo”中查询所有性别为“男”的学生所有信息。

(2) SELECT 学号，姓名 FROM StuInfo WHERE 入学成绩 > = 530

在学生信息表“StuInfo”中查询所有入学成绩大于等于 530 分的学生学号和姓名。

2. CREATE 语句

CREATE 语句用来在数据库中建立新的数据表。其基本语法格式为：

CREATE TABLE 表名
([字段名 1] 数据类型 1(长度 1),
[字段名 2] 数据类型 2(长度 2),
… … … … … …)

例如，创建一个表名为 TestTable，两个字段分别为“Field1”和“Field2”的数据库表：

CREATE TABLE TestTable (Field1 Text(8), Field2 Text(16))

3. INSERT 语句

INSERT 语句用来在指定的数据库表中插入新的记录。其基本语法格式为：

INSERT INTO 表名 [字段名表] VALUES(值表)

说明：

(1) “表名”用来指定要进行插入的数据库表。

(2) “字段名表”及“值表”用来指定插入字段的值。

例如，在学生信息表中插入一学号为“8060306”，姓名为“王宏亮”的记录：

INSERT INTO StuInfo 学号，姓名 VALUES('08060306','王宏亮')

4. DELETE 语句

DELETE 语句用来在指定的数据库表中删除符合条件的记录。其基本语法格式为：

DELETE FROM 表名 [WHERE 条件]

例如，在学生信息表中删除入学成绩小于 500 分的记录：

DELETE FROM StuInfo WHERE 入学成绩＜500

5. UPDATE 语句

UPDATE 语句用来在指定的数据库表中修改记录。其基本语法格式为：

UPDATE SET 字段名 1＝表达式 1［,字段名 2＝表达式 2 ］［,…］［WHERE 条件］

例如，将学生名为“王宏亮”记录的性别修改为“男”，成绩修改为 518：

UPDATE SET 性别＝'男',入学成绩＝518 WHERE 姓名＝'王宏亮'

10.2　可视化数据管理器

除了可以使用相关数据库管理系统（如 Access、SQL Server 等）来创建及管理数据库外，在 Visual Basic 中，还可以利用可视化数据管理器（Visual Data Manager）直接创建及管理数据库。可视化数据管理器是一个非常方便的数据操作工具，利用它可以方便地建立数据库、数据库表以及对数据库表进行修改、添加、删除和查询等操作。

10.2.1　启动可视化数据管理器

在 Visual Basic 集成开发环境中，单击“外接程序”菜单的“可视化数据管理器”命令，即可打开可视化数据管理器（VisData），其窗口如图 10－1 所示。

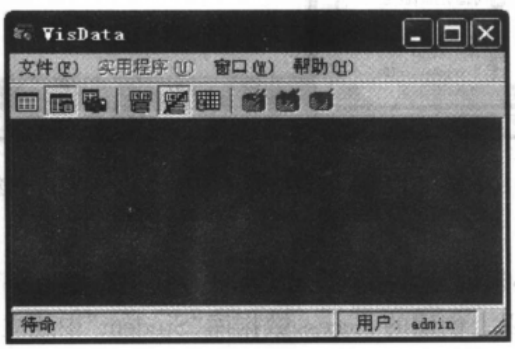


图 10－1　可视化数据管理器窗口

10.2.2　利用可视化数据管理器创建及管理数据库

1. 创建数据库

创建数据库的操作步骤如下：

（1）选择“文件”菜单中的“新建”命令，在子菜单中选择要建立的数据库类型。如选择 Microsoft Access（Access 数据库），再从其子菜单中选择 Version 7.0 MDB（7）命令，如图 10－2 所示。

（2）在打开的对话框中选择要建立的数据库所在的文件夹并输入数据库文件名。如输入数据库文件名为 Student。

（3）单击对话框的“保存”按钮保存数据库后，VisData 窗口将打开两个子窗口。在左边的数据库窗口，单击“Properties”项左边的“＋”标记，可以看到数据库的常用属性。右边为 SQL 语句窗口，在此窗口可以输入 SQL 语句对数据库进行操作。如图 10－3 所示。

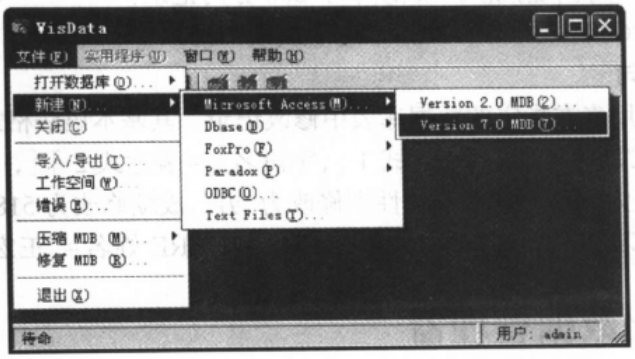


图 10－2　利用可视化数据管理器新建数据库

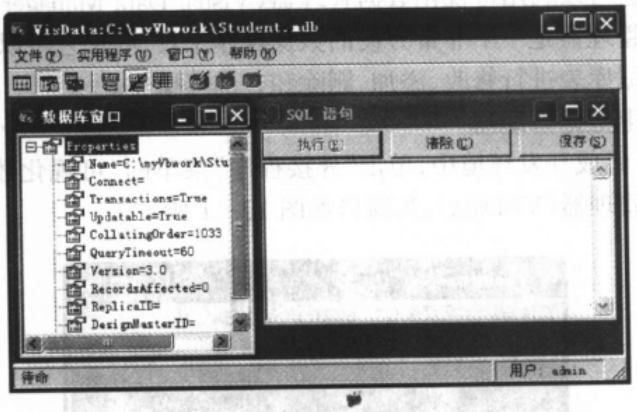


图 10－3　VisData 的数据库子窗口与 SQL 语句子窗口

2. 建立数据库表

建立数据库后，还必须创建数据库表，因为表是真正存储数据的地方，没有表的数据库就无法存储数据。如在以上创建的 Access 数据库“Student”中建立一个“StuInfo”表，可用如下步骤完成：

(1) 在 VisData 的窗口中，右键单击数据库窗口，从弹出的快捷菜单中选择“新建表”命令，则打开如图 10－4 所示的“表结构”对话框。

利用“表结构”对话框，可以建立数据库表的结构(字段和名称、字段的类型及字段的大小等)。“表结构”对话框中与表结构相关的各项内容及其说明见表 10－2。

表 10－2　“表结构”对话框中表结构的相关项目及其说明

项　目	说　明
表名称	用来输入要建立表的名称
字段列表	显示当前表中已包含的字段名
名称	用来显示或修改当前在字段列表中选择的字段名
类型	显示当前在字段列表中选择字段的类型
大小	显示当前在字段列表中选择字段的最大长度(以字节为单位)
顺序位置	确定字段的相对位置

续表 10－2

项　目	说　明
验证文本	用户输入的字段值无效时，应用程序将显示消息文本
验证规则	用来确定字段取什么样数据值，如“＞＝0 And ＜＝100”，表示字段值为 0～00
缺省值	用来指定该字段的默认值
“添加字段”按钮	单击该按钮，将显示“添加字段”对话框，用来添加新的字段
“删除字段”按钮	单击该按钮，将删除当前在字段列表框中选中的字段

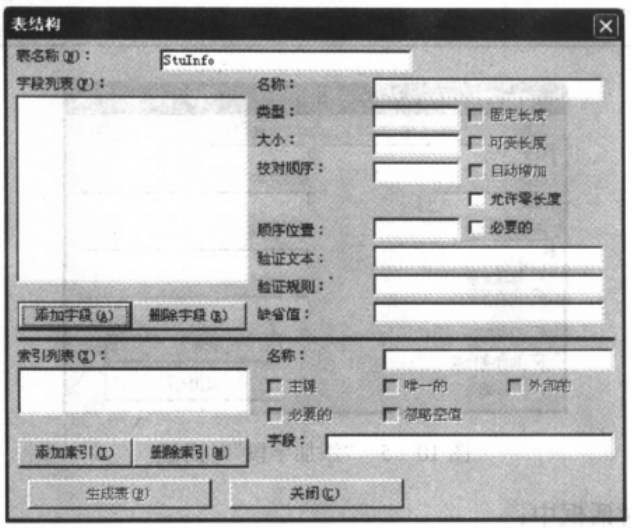


图 10－4　“表结构”对话框

“表结构”对话框的下半部分用于设置或显示当前表的索引信息。其各项内容的含义见表 10－3。

10－3　“表结构”对话框中索引的相关项目及其说明

项　目	说　明
索引列表	用来显示当前已经建立的索引
名称	用来显示或修改索引名
主建	选中时表示当前索引为表的主索引
惟一的	选中时表示当前索引字段应具有惟一的值
外部的	选中时表示当前索引字段是表的外部键
必要的	选中时表示当前索引必须是非空(Null)值
忽略空值	选中时表示含有 Null 值的字段不包括在索引之中
“添加索引”按钮	单击该按钮，显示“添加索引”对话框，用来添加新的索引
“删除索引”按钮	单击该按钮，删除当前在索引列表框中选中的索引

(2) 在“表结构”对话框中，输入要创建的数据库表名，如“StuInfo”。

(3) 单击“添加字段”按钮，打开 “添加字段”对话框。在“添加字段”对话框中设置要添加的字段信息(字段名称、类型、长度等)，StuInfo 数据库表的各字段的信息见表 10－4。如

字段名输入为“学号”,字段类型选择“Text”类型,字段长度设置为 8(图 10－5)。单击“确定”按钮后,“表结构”对话框中即显示添加的字段信息。

表 10－4 StuInfo 数据库表的字段

字段名称	字段类型	字段长度
学号	Text	8
姓名	Text	10
性别	Text	2
出生日期	Data/Time	8(自动设置)
家庭地址	Text	50
入学成绩	Integer	2(自动设置)

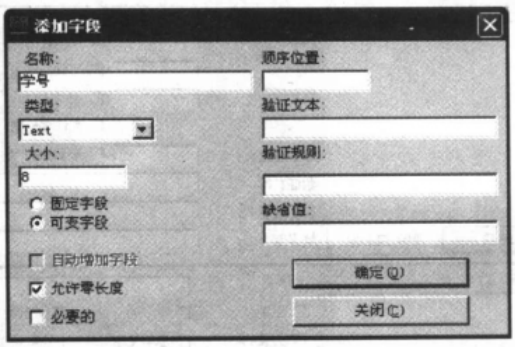


图 10－5 “添加字段”对话框

在“添加字段”对话框中:

① “固定字段”和“可变字段” 单选按钮只对 Text 类型的字段起作用。“固定字段”选定时用来表示当前的字段长度是固定的,“可变字段”选定时用来表示当前的字段长度是可变的。

② “自动增加字段”复选框,只对 Long 类型的字段起作用。若选定,则当向表中添加记录时,该字段的值会在上一条记录的基础上自动增加 1。

③ 其余各项与“表结构”对话框相同。

(4) 各字段添加完成后,在“表结构”对话框中单击“添加索引”按钮,利用打开的“添加索引”对话框(图 10－6)为数据库表 StuInfo 添加索引。如在“可用字段”列表框中选择“学号”字段,在名称中输入“StuID”作为索引名称(索引名称也可以与选用的字段名相同),并作

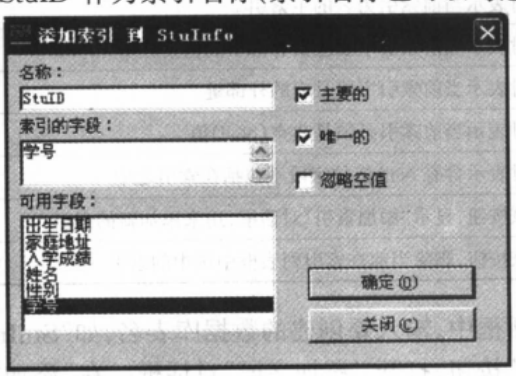


图 10－6 “添加索引”对话框

为主要的和惟一的索引，单击“确定”按钮后则在“表结构”对话框中显示表索引信息。

(5) 添加完字段和索引后，在“表结构”对话框中单击“生成表”按钮后，就在数据库中添加了一个新表。此时在 VisData 的“数据库窗口”中即可显示所添加的数据库表的字段、索引及属性等信息，如图 10－7 所示。

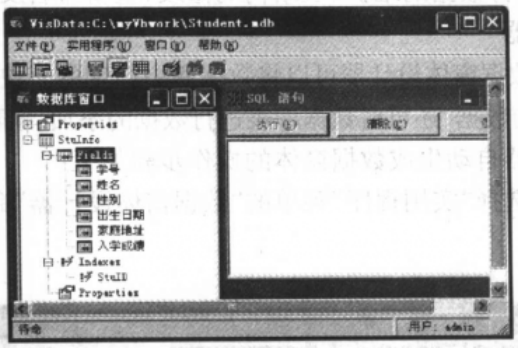


图 10－7　在“数据库窗口”中显示的数据表信息

利用以上方法，还可以在同一个数据库中添加其他表(如学生成绩表等)。

3. 数据库表的维护

利用以上方法建立数据库表后，此时所建的数据库表为一空表，也就是说仅仅是建立了表的结构，表中并没有数据。

在 VisData 的“数据库窗口”中，右键单击数据库表选择“打开”命令或双击数据库表，则弹出如图 10－8 所示的窗口。利用此窗口可以对数据库表进行数据的添加、删除及修改等维护操作。例如，单击“添加”按钮可向数据库表中添加新记录。

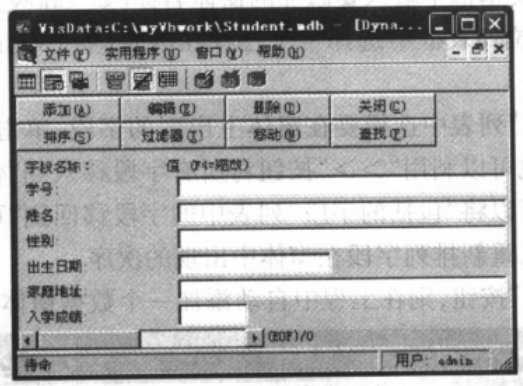


图 10－8　数据库表的维护窗口

4. 打开数据库

在 VisData 中，可以利用“文件”菜单中的“打开数据库”命令来打开一个已经存在的数据库。

例如，打开上面创建的 Access 数据库 Student，则：

(1) 选择“文件”菜单中的“打开数据库”命令，在子菜单中选择 Microsoft Access，此时将

显示“打开 Microsoft Access 数据库”对话框。

(2) 在“打开 Microsoft Access 数据库”对话框中,在相应的路径下选择“Student”数据库文件,单击“打开”按钮,即可将数据库 Student 数据库打开。

将数据库打开后,右键单击相应的数据库表,选择“打开”、“结构”、“重命名”及“删除”等命令,可以对数据库表进行数据的维护、表结构的修改、重新命名及删除表等操作。

10.2.3 数据窗体设计器

在 VisData 中,利用数据窗体设计器可以将数据库中指定的表快速生成一个窗体,通过此窗体可以显示数据库表的数据,还可对数据库表进行数据的添加、删除、修改等维护操作。

利用数据窗体设计器自动生成数据窗体的操作步骤如下:

(1) 在 VisData 中,选择“实用程序”菜单的“数据窗体设计器”命令,打开“数据窗体设计器”对话框(图 10-9)。

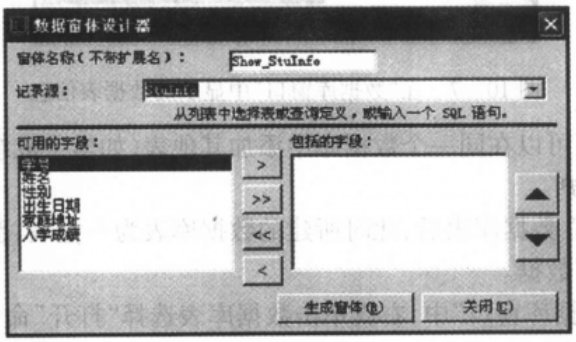


图 10-9 “数据窗体设计器”对话框

(2) 在“窗体名称”文本框中输入要添加的窗体名称(如“Show_StuInfo”)。

(3) 在“记录源”下拉列表框中选择一个已存在的表(如“StuInfo”)或输入一个有效的 SQL 语句。

(4) 在“可用的字段”列表中选择要在窗体上出现的字段,单击“>”按钮将该字段移到“包括的字段”列表中,也可以利用“>>”按钮将所有字段移入到“包括的字段”列表中。同样,使用“<”或“<<”可以将“包括的字段”列表中的字段移回到“可用的字段”列表中。利用右侧的上、下箭头可以重新排列字段在窗体中出现的次序。

(5) 单击“生成窗体”按钮,则在工程中自动添加一个数据窗体(图 10-10)。该窗体名

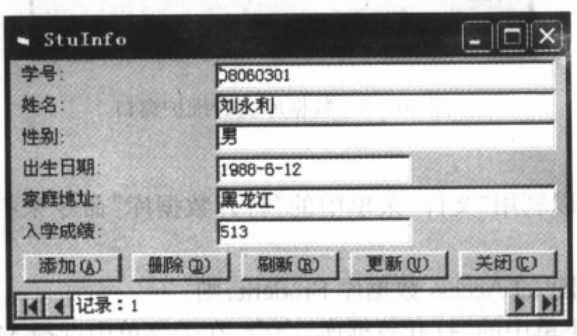


图 10-10 利用“数据窗体设计器”自动生成数据窗体

为在“数据窗体设计器”中输入的窗体名称前加上“frm”，如“frmShow_StuInfo”。

10.3　数据库控件

如果要通过 Visual Basic 操作数据库，主要涉及到两大类数据库控件，即数据库访问控件和数据显示控件。

数据库访问控件用于建立数据库连接，提供对数据的访问。Visual Basic 有 3 种控制连接和访问数据库的对象，分别是 DAO(数据访问对象)、RDO(远程数据访问对象)和 ADO(ActiveX 数据对象)。其中 ADO 是 Microsoft 公司最新、功能最强大的数据访问对象，可以用来访问各种数据源。而 Adodc 控件是集成了 ADO 对象基本功能的常用数据库访问控件。

数据显示控件(数据绑定控件)用于显示数据库表中的内容。使用数据库访问控件只能实现与数据源的连接，并返回所需要的记录集，但它没有提供显示数据的功能。要显示由数据访问控件所获得的数据集中的数据，就必须使用数据显示控件。

10.3.1　Adodc 控件

Adodc 控件是集成了 ADO 对象基本功能的常用数据库访问控件，使用该控件可以不用编写许多代码，就可以直接完成许多操作。

1. Adodc 控件的添加

Adodc 控件不是 Visual Basic 的标准控件，因此在使用前必须将其添加到工具箱中。将 Adodc 控件添加到工具箱的方法是：选择“工程”菜单中的“部件”命令(或右键单击工具箱，选择“部件”命令)，在弹出的“部件”对话框的“控件”选项卡中，选择部件列表中的 Microsoft ADO Data Control 6.0(OLEDB)选项，单击“确定”按钮退出对话框。则 Adodc 控件就添加到工具箱中，其在工具箱中的图标如图 10－11 所示。这时就可以像添加其他普通控件一样将其添加到窗体上，其外观如图 10－12 所示。

图 10－11　工具箱中的 Adodc 控件

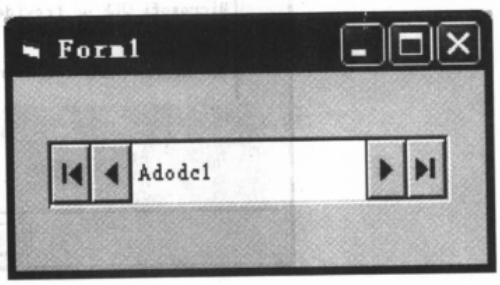


图 10－12　Adodc 控件的外观

2. 利用 Adodc 控件连接数据库

在 Adodc 控件中，利用其 ConnectionString 属性可以快速地建立和数据库的连接，而其 RecordSource 属性则用来确定具体可以访问的记录集。利用 Adodc 控件连接数据库及确定访问的数据集的步骤如下：

(1) 在窗体中放置 Adodc 控件，其默认名为 Adodc1。

(2) 设置 Adodc1 控件的 ConnectionString 属性，实现与数据库的连接。其操作步骤如下：

① 在“属性”窗口中用鼠标单击 Adodc1 控件的 ConnectionString 属性右边的省略号按钮，弹出 Adodc 控件的“属性页”对话框，如图 10－13 所示。

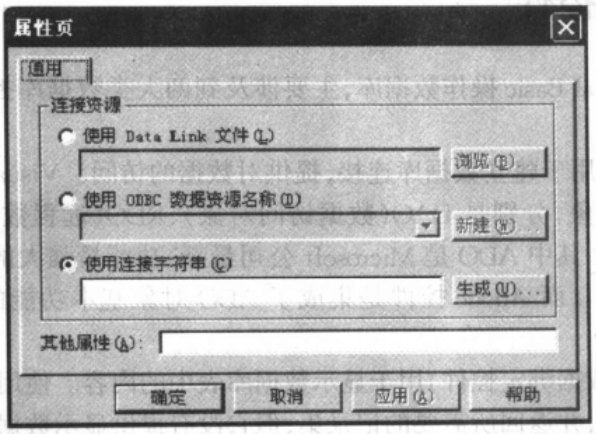


图 10－13 Adodc 控件的“属性页”对话框

② 在 Adodc 控件的“属性页”对话框中选择“使用连接字符串”，单击“生成”按钮，弹出“数据链接属性”对话框，如图 10－14 所示。

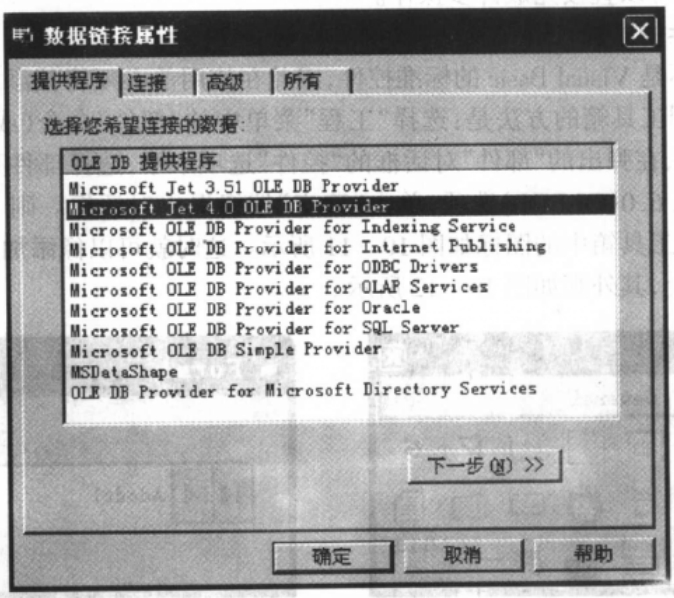


图 10－14 “数据链接属性”对话框的“提供程序”选项卡

③ 在“提供程序”选项卡中选择“Microsoft Jet 4.0 OLE DB Provider”，表示要连接一个 Access 数据库(其余选项用来连接相应的数据库，如选择“Microsoft OLE DB Provider for SQL Server”，则可以连接一个 SQL Server 数据库)。单击“下一步”按钮，打开“连接”选项卡，如图 10－15 所示。

④ 单击“选择或输入数据库名称”项右侧的省略号按钮，利用打开的“选择 Access 数据库”对话框选择一个已存在的 Access 数据库，如“Student”数据库。然后单击“连接”选项卡

中的“测试连接”按钮，如果数据库连接成功，则会出现一个“测试连接成功”的提示。再单击“数据链接属性”对话框的“确定”按钮，则在“属性页”对话框中生成连接字符串。

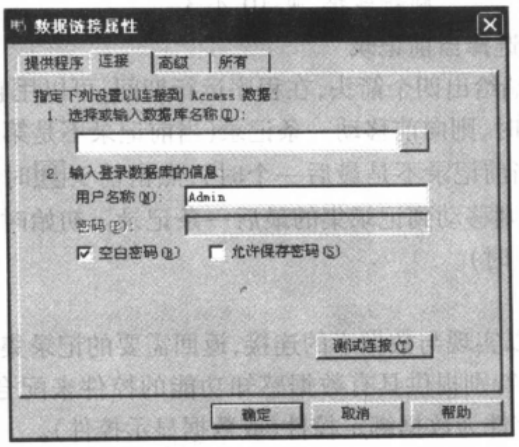


图 10－15　“数据链接属性”对话框的“连接”选项卡

⑤ 最后，单击“属性页”对话框的“确定”按钮，完成了 Adodc 控件的 ConnectionString 属性设置。即通过 Adodc 控件的 ConnectionString 属性实现了与数据库的连接。

(3) 设置 Adodc 控件的 RecordSource 属性，确定访问的数据集。

在 Adodc 控件的属性窗口中，单击 RecordSource 属性右侧的省略号按钮，弹出 RecordSource 属性的“属性页”对话框，如图 10－16 所示。利用该对话框来设置 RecordSource 属性，确定要访问的数据集。

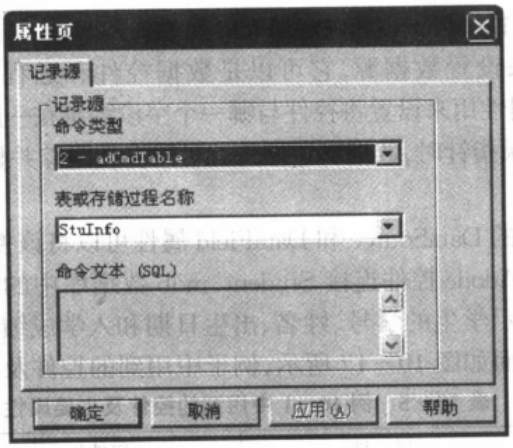


图 10－16　RecordSource 属性的“属性页”对话框

RecordSource 属性的值可以是一个表名称、一个存储查询或一个查询字符串。

例如，设置“记录源”的“命令类型”为“2 - adCmdTable”，即要访问的数据集为一数据库表；在“表或存储过程名称”组合框中选择要访问的表，如“StuInfo”。

单击“确定”按钮，即可完成 RecordSource 属性的设置，确定了要访问的记录集。

除了在程序设计阶段通过属性窗口对 Adodc 控件的 ConnectionString 和 RecordSource 属性进行设置来实现与数据库的连接外，还可以在程序运行时期通过对这两个属性赋值的

方式来实现与数据库的动态连接。例如：

```
Adodc1.RecordSource="Select * From StuInfo Where 入学成绩>=520"
Adodc1.Refresh        '刷新连接,见 10.4.3
```

3.利用 Adodc 控件选择当前记录

Adodc 控件上的两端给出四个箭头，在程序运行期间，可以用它们来选择记录集中当前的记录。当点击箭头◀时，则向前移动一条记录(当前记录不是第一个时)；点击箭头▶时，则向后移动一条记录(当前记录不是最后一个时)；点击箭头⏮时，则移动到记录集的第一条记录；点击箭头⏭时，则移动到记录集的最后一条记录。初始时，当前记录为记录集中第一条记录(若记录集不空时)。

10.3.2 数据绑定控件

Adodc 控件虽然可以实现与数据库的连接，返回需要的记录集，但它并没有提供显示数据的功能。Visual Basic 特别提供具有数据感知功能的控件来配合 Adodc 控件显示记录集中的数据内容，称这些控件为数据绑定控件(或数据显示控件)。

任何具有 DataSource 属性的控件都可绑定到 Adodc 控件上。可以绑定到 Adodc 控件上的标准控件有：文本框(TextBox)、复选框(CheckBox)、组合框(ComboBox)、图像框(Image)、标签(Label)、列表框(ListBox)以及图片框(PictureBox)等控件；可绑定到 Adodc 控件上的 ActiveX 控件有：DataGrid、DataList、DataCombo、RichTextBox、ImageCombo、Microsoft Chart、MonthView 以及 DataTimerPicker 等控件。

在数据绑定控件中大都有两个共同属性，即 DataSource 和 DataField 属性。下面介绍数据绑定控件的这两个主要属性，并以 TextBox 和 DataGrid 控件为例介绍数据绑定控件的使用方法。

1.数据绑定控件的 DataSource 和 DataField 属性

DataSource 属性用来设置数据源，它可以是数据控件和远程数据控件，当然也可以是 Adodc 控件。DataField 属性用来设置将控件与哪一个字段绑定在一起。有了这两个属性后，在使用 Adodc 控件移动记录指针时，数据绑定控件就可以自动跟踪并显示当前记录的数据。

2.TextBox 控件

利用 TextBox 控件的 DataSource 和 DataField 属性可以将该控件绑定到 Adodc 控件上。

【例 10-1】 使用 Adodc 控件连接 Student.mdb 数据库中的学生基本信息表 StuInfo，并通过 TextBox 控件显示学生的学号、姓名、出生日期和入学成绩。

本例的程序运行界面如图 10-17 所示，例子中用到的控件及主要属性见表 10-5。

表 10-5 例 10-1 中用到的控件及主要属性

对 象	属 性	属性值	说 明
Form1	Caption	学生基本信息	Form 控件
Label1	Caption	学号：	Label 控件
Label2	Caption	姓名：	Label 控件
Label3	Caption	出生日期：	Label 控件
Label4	Caption	入学成绩：	Label 控件
Adodc1	Caption	学生基本信息表	Adodc 控件

续表 10－5

对　象	属　性	属性值	说　明
TextID	DataSource	Adodc1	TextBox 控件，用来显示学号
	DataField	学号	
TextName	DataSource	Adodc1	TextBox 控件，用来显示姓名
	DataField	姓名	
TextBirthday	DataSource	Adodc1	TextBox 控件，用来显示出生日期
	DataField	出生日期	
TextScore	DataSource	Adodc1	TextBox 控件，用来显示入学成绩
	DataField	入学成绩	

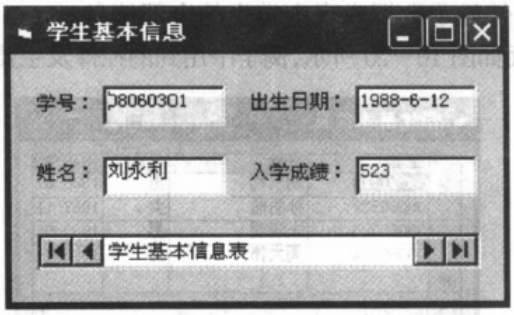


图 10－17　例 10－1 程序运行界面

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 在 Form1 窗体上依次添加表 10－5 中所示各个控件。

(3) 设置 Adodc1 控件的 ConnectionString 和 RecordSource 属性，将其连接到“Student.mdb”数据库中的“StuInfo”表。具体方法参见 10.3.1。

(4) 设置其他控件的属性(见表 10－5)。

3. DataGrid 控件

DataGrid 控件是一个可以绑定到 Adodc 控件上的 ActiveX 控件，该控件能够以表格的形式显示记录集中的记录。

(1) DataGrid 控件的添加。在使用该控件前要先将该控件添加到工具箱中。将 DataGrid 控件添加到工具箱的方法是：选择“工程”菜单中的“部件”命令(或右键单击工具箱，选择“部件”命令)，在弹出的“部件”对话框中的“控件”选项卡中，选择部件列表中 Microsoft DataGrid Control 6.0(OLEDB)选项，单击“确定”按钮退出对话框。则 DataGrid 控件就添加到工具箱中，其在工具箱中的图标如图 10－18 所示。添加到窗体上的 DataGrid 控件的外观如图 10－19 所示。

(2) DataGrid 控件的属性。DataGrid 控件常用的主要属性有：

① DataSource 属性：与 TextBox 控件的 DataSource 相同。可用于绑定到 Adodc 控件，指定要显示的记录集。

② AllowAddNew 属性：属性值为 Boolean 类型，用于设置此控件是否可以添加记录。

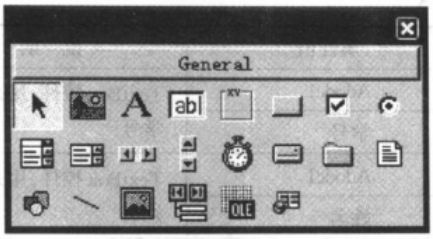


图 10－18　工具箱中的 DataCird 控件

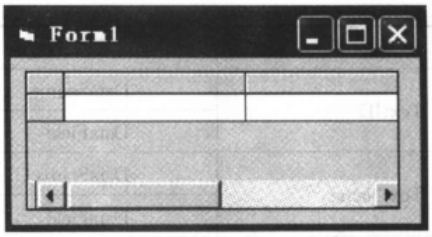


图 10－19　DataCird 控件的外观

③ AllowDelete 属性：属性值为 Boolean 类型，用于设置此控件是否可以删除记录。

④ AllowUpdate 属性：属性值为 Boolean 类型，用于设置此控件是否可以修改记录。

【例 10－2】 使用 Adodc 控件连接 Student.mdb 数据库中的 StuInfo 表(学生基本信息表)，并通过 DataGrid 控件显示数据库表中学生的全部信息。

本例的程序运行界面如图 10－20 所示，例子中用到的控件及主要属性见表 10－6 所示。

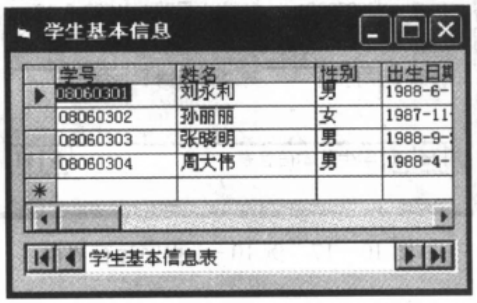


图 10－20　例 10－2 程序运行界面

表 10－6　例 10－2 中用到的控件及主要属性

对　象	属　性	属性值	说　明
Form1	Caption	学生基本信息	Form 控件
Adodc1	Caption	学生基本信息表	Adodc 控件
DataGrid1	DataSource	Adodc1	DataGrid 控件
	AllowAddNew	True	
	AllowDelete	True	
	AllowUpdate	True	

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 在 Form1 窗体上依次添加一个 Adodc 控件和一个 DataGrid 控件。

(3) 设置 Adodc1 控件的 ConnectionString 和 RecordSource 属性，将其连接到“Student.mdb”数据库中的“StuInfo”表。

(4) 设置 DataGrid 控件的属性见表 10－6。

在本例中，当程序运行时，可以通过 DataGrid 控件的最后一个空行向数据集中添加新的记录；也可以选定一条记录(鼠标单击记录左侧)，按下键盘上的“Del”键删除该记录；还可

以直接修改记录中的各字段值。若将 DataGrid 控件的 AllowAddNew、AllowDelete 及 AllowUpdate 属性值设为 False，则此时就不能对所显示的数据进行编辑，包括添加和删除。当然也可以在程序运行时指定这些属性的值。

10.4　Adodc 控件的高级成员

对于 Adodc 控件，除了前面介绍的 ConnectionString 属性和 RecordSource 属性用来连接数据库和获得要访问的数据集之外，Adodc 控件还有一些常用的属性和方法。要使用 Adodc 控件实现一个较为复杂的信息管理系统，则必须掌握 Adodc 控件的高级成员的使用方法。

10.4.1　CommandType 属性

Adodc 控件的 CommandType 属性用来指定 RecordSource 属性的类型。在程序设计时可以直接在 Adodc 控件的属性窗口中 CommandType 属性框右边的下拉列表中选择一种类型，也可以在程序运行期间动态地对 CommandType 属性进行设置。表 10－7 给出 CommandType 属性的可用值及其说明。

表 10－7　CommandType 属性的可用值

属性值	属性常量	说　明
8	AdCmdUnknown	缺省值，表明 RecordSource 为 SQL 语句
1	AdCmdText	表明 RecordSource 为 SQL 语句
2	AdCmdTable	表明 RecordSource 为数据库表名
4	AdCmdStoreProc	表示 RecordSource 为存储过程调用

10.4.2　RecordSet 属性

RecordSet 属性表示由记录组成的集合。RecordSet 属性为一对象，其成员属性和方法用于对记录集进行操作。下面介绍其主要属性和方法。

1. Fields 属性

RecordSet 对象的 Fields 属性可用于对记录集的字段进行操作。Fields 属性也是一个对象，其属性用来对字段进行操作。

(1) 利用 Fields 属性表示字段的方法。利用 Fields 属性可以采用名称法和下标法来表示记录集中的字段。例如，表示 StuInfo 表中的“姓名”字段：

① 名称法：Fields("姓名")。

② 下标法：Fields(1)

使用下标法表示字段时，其下标从 0 开始。即第一个字段下标为 0，第二个字段下标为 1，依次类推。

(2) Count 属性。Count 属性可用来获取记录集中的字段个数。例如：

```
Adodcl.RecordSet.Fields.Count        '获得 Adodcl 连接的记录集中字段的个数
```

(3) Name 属性。Name 属性可用来获取某个字段的字段名。例如：

```
Adodcl.RecordSet.Fields(1).Name      '获取记录集中的第二个字段的字段名
```

(4) Value 属性。Value 属性可用来获取或设置记录集中某个字段的值。它也是 Fields

对象默认的属性,可以省略。例如:

```
Adodcl.RecordSet.Fields("入学成绩").Value =534
```

则表示将当前记录的"入学成绩"字段值设置为 534。其等价于以下语句为:

```
Adodcl.RecordSet.Fields("入学成绩") =534
```

2.EOF 和 BOF 属性

这两个属性用于判断当前记录指针是否超出记录集头部或尾部。如果当前记录已经位于第 1 条记录时,再调用 MovePrevious 方法向前移动记录指针时,RecordSet 对象的 BOF 属性会被置为 True,其余情况则 BOF 属性值为 False。如果当前记录已经位于最后一条记录,再调用 MoveNext 方法向后移动记录指针时,RecordSet 对象的 EOF 属性会被置为 True,否则其值为 False。

3.Move 方法组

Move 方法组用来移动记录指针,改变当前记录。共有五个成员:

(1) MovePrevious 方法:将记录指针移到当前记录的前一条记录。

(2) MoveNext 方法:将记录指针移到当前记录的下一条记录。

(3) MoveFirst 方法:将记录指针移到第一条记录。

(4) MoveLast 方法:将记录指针移到最后一条记录。

(5) Move:移动到指定位置的记录。例如:

```
Adodc1.RecordSet.Move 4      '向后移动 4 条记录
```

4.数据操作成员

RecordSet 对象还提供了三个可以用来实现数据的添加、删除和修改的方法,即 AddNew、Delete 和 Update。

(1) AddNew 方法。AddNew 方法用于在记录集尾端添加一条新记录(空记录)。例如:

```
Adodc1.RecordSet.Addnew
```

即可实现在记录集的尾端添加一条新的空白记录。此后还应该给此记录各个字段赋以相应的值,再调用 Update 方法保存记录。

(2) Delete 方法。Delete 方法用来删除当前记录。例如:

```
Adodc1.RecordSet.Addnew
```

即可将当前记录删除。

(3) Update 方法。用于保存添加的新记录或记录修改后的内容。例如:

```
Adodc1.RecordSet.Addnew
Adodc1.RecordSet.Fields(0) ="08060310"
Adodc1.RecordSet.Update
```

10.4.3 Refresh 方法

Refresh 方法用于刷新 Adodc 控件对数据库的连接。当在程序运行期间,对 Adodc 控件的 ConnectionString 和 RecordSource 属性重新设置后,需要利用 Refresh 方法来刷新对数据库的连接及获取新的记录源。例如:

```
Adodc1.CommandType=8      '设置 RecordSource 为 SQL 语句
Adodcl.RecordSource ="Select 学号,姓名,入学成绩 from StuInfo"
Adodc1.Refresh
```

10.5　应用实例

【例 10－3】 利用本章建立的学生数据库(Student.mdb)中的学生基本信息表(StuInfo)编写一个学生基本信息管理程序。

该程序实现如下功能：

(1) 利用 Adodc 控件连接数据库，并通过 TextBox 和 DataGrid 控件对学生信息进行浏览显示。

(2) 能够实现学生信息的添加、删除、修改及查询等功能。

本实例中用到了两个窗体，主窗体(图 10－21)用来实现对学生信息的浏览、添加、修改及删除等操作，以及浏览查询结果等。“选择查询条件”窗体(图 10－22)用来实现在查询时选择查询条件。主窗体显示的查询结果如图 10－23 所示。

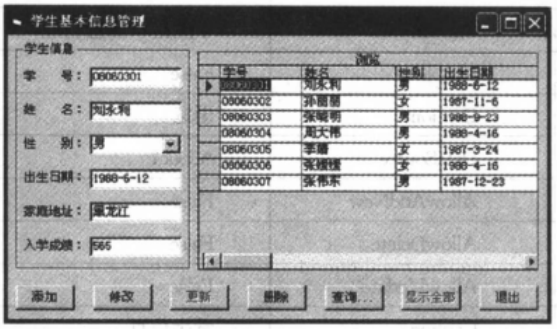


图 10－21　“学生基本信息管理程序”主界面

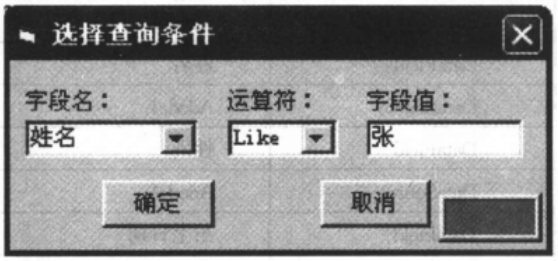


图 10－22　“选择查询条件”界面

实例中用到的各个控件及其主要属性见表 10－7 和表 10－8 所示，其中对于 Label 控件不赘述。

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 在 Form1 窗体上添加一个 Adodc 控件并连接到 Student.mdb 数据库中的 Stuinfo 表。其中设置 RecordSource 属性时，选择类型为 8－AdCmdUnknown，输入的 SQL 语句为“Select * From StuInfo”。

(3) 在 Form1 窗体上依次添加其他各控件，并进行属性设置。

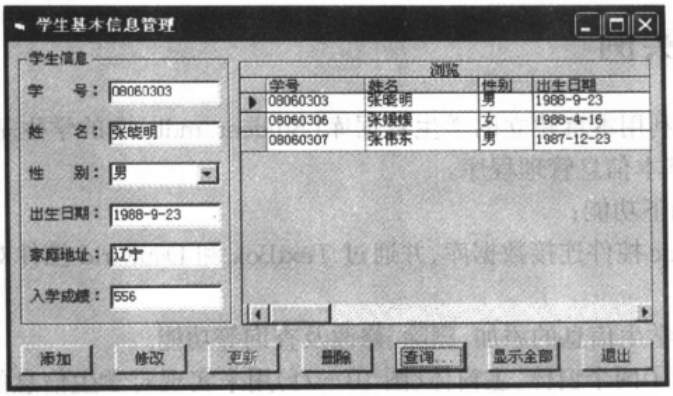


图 10－23　查询结果界面

表 10－7　主界面的控件及主要属性

对　象	属　性	属性值	说　明
Form1	Caption	学生基本信息管理	Form 控件
Adodc1	Visible	False	Adodc 控件
DataGrid1	Caption	浏览	DataGrid 控件
	DataSource	Adodc1	
	AllowAddNew	False	
	AllowDelete	False	
	AllowUpdate	False	
Frame1	Caption	学生信息	
TextID	DataSource	Adodc1	TextBox 控件，用来显示学号
	DataField	学号	
TextName	DataSource	Adodc1	TextBox 控件，用来显示姓名
	DataField	姓名	
ComboSex	DataSource	Adodc1	TextBox 控件，用来显示性别
	DataField	性别	
TextBirthday	DataSource	Adodc1	TextBox 控件，用来显示出生日期
	DataField	出生日期	
TextAddr	DataSource	Adodc1	TextBox 控件，用来显示家庭地址
	DataField	家庭地址	
TextScore	DataSource	Adodc1	TextBox 控件，用来显示入学成绩
	DataField	入学成绩	
CmdAdd	Caption	添加	CommandBotton 控件
CmdEdit	Caption	修改	CommandBotton 控件
CmdUpdate	Caption	更新	CommandBotton 控件
CmdDel	Caption	删除	CommandBotton 控件
CmdSearch	Caption	查询	CommandBotton 控件

续表 10－7

对　象	属　性	属性值	说　明
CmdShowAll	Caption	显示全部	CommandBotton 控件
CmdExit	Caption	退出	CommandBotton 控件

表 10－8　“选择查询条件”界面的控件及主要属性

对　象	属　性	属性值	说　明
Form2	Caption	选择查询条件	Form 控件
	BorderStyle	3－Fixed Dialog	
Combo1			ComboBox 控件
Combo2			ComboBox 控件
Text1	Text	(空)	Text 控件
OkButton	Caption	确定	CommandBotton 控件
CancelButton	Caption	取消	CommandBotton 控件

(4) 在当前工程中新添一个窗体(Form2),用于查询时选择查询条件。并向该窗体中添加相应的控件(表 10－8)并设置其属性。

(5) 在当前工程中添加一个模块,并命名为 MouduleSearch.bas。并编写其代码(见代码部分)。

(6) 编写 Form1 的程序代码(见代码部分)。

(7) 编写 Form2 的程序代码(见代码部分)。

程序中编写的代码如下:

(1) 在 ModuleSearch.bas 模块中定义如下全局变量:

```
Global SearchString As String      '用于获取查询条件
Global SearchFlag As Integer       '用于获取是否进行查询的标识
```

(2) Form1 窗体代码:

```
Private Sub InitShow()             '在通用程序段内定义,用于主界面属性设置
    TextID.Locked = True           '只用于显示,不能编辑。下同
    TextName.Locked = True:ComboSex.Locked = True
    TextBirthday.Locked = True:TextAddr.Locked = True
    TextScore.Locked = True
    CmdAdd.Enabled = True:CmdDel.Enabled = True     '设置控件是否可用。下同
    CmdEdit.Enabled = True:CmdSearch.Enabled = True
    CmdUpdate.Enabled = False:CmdShowAll.Enabled = False
End Sub
Private Sub Form_Load()            '窗体的 Load 事件
    CmdUpdate.Enabled = False
    ComboSex.AddItem "男"
    ComboSex.AddItem "女"
```

```
    InitShow
End Sub
Private Sub CmdAdd_Click()              '"添加"按钮的 Click 事件
    '设置 TextBox 及 ComboBox 控件可以编辑
    TextID. Locked = False: TextName. Locked = False
    ComboSex. Locked = False: TextBirthday. Locked = False
    TextAddr. Locked = False: TextScore. Locked = False
    CmdAdd. Enabled = False             '设置控件是否可用,下同
    CmdDel. Enabled = False: CmdEdit. Enabled = False
    CmdSearch. Enabled = False: CmdUpdate. Enabled = True
    Adodc1. Recordset. AddNew           '添加新记录
End Sub
Private Sub CmdEdit_Click()             '"修改"按钮的 Click 事件
    CmdAdd. Enabled = False: CmdDel. Enabled = False
    CmdEdit. Enabled = False: CmdSearch. Enabled = False
    CmdUpdate. Enabled = True
    TextID. Locked = False: TextName. Locked = False
    ComboSex. Locked = False: TextBirthday. Locked = False
    TextAddr. Locked = False: TextScore. Locked = False
End Sub
Private Sub CmdUpdate_Click()
    '"更新"按钮的 Click 事件,用于记录添加、修改后重新显示记录信息
    Adodc1. Recordset. Update
    Adodc1. Recordset. MoveFirst
    InitShow
End Sub
Private Sub CmdDel_Click()              '"删除"按钮的 Click 事件
    Dim DelFlag As Integer
    DelFlag = MsgBox("是否要删除此记录", vbOKCancel + vbQuestion,"提示")
    If DelFlag = 1 Then                 '当确定要进行删除时
        Adodc1. Recordset. Delete
    End If
End Sub
Private Sub CmdSearch_Click()           '"查询"按钮的 Click 事件
    Form2. Show 1                       '显示"选择查询条件"窗体,指定查询条件
    If SearchFlag = 1 Then              '确定要进行查询时
        Adodc1. RecordSource = "select * from stuinfo where " & SearchString
        Adodc1. Refresh
        CmdShowAll. Enabled = True      '全部显示'按钮可用
```

```
    End If
End Sub
Private Sub CmdShowAll_Click()                    '"全部显示"按钮的 Click 事件
    Adodc1.RecordSource="select * from stuinfo"
    Adodc1.Refresh
    InitShow
End Sub
Private Sub CmdExit_Click()                       '"退出"按钮的 Click 事件
    Unload Me
End Sub
```

(3) Form2 窗体的代码

```
Private Sub Form_Load()                           'Form2 窗体的 Load 事件
    Dim i As Integer
    Combo1.Clear                                  '清空"字段"组合框中的选项
    For i=0 To Form1.Adodc1.Recordset.Fields.Count-1
        Combo1.AddItem Form1.Adodc1.Recordset.Fields(i).Name
    Next i                    '通过循环将学生信息表中各字段名加至"字段"组合框
    Combo2.Clear              '清空"运算符"组合框中的选项,并添加如下运算符
    Combo2.AddItem "=":Combo2.AddItem ">":Combo2.AddItem ">="
    Combo2.AddItem "<":Combo2.AddItem "<=":Combo2.AddItem "<>"
    Combo2.AddItem "Like"                         '用于糊模查询
End Sub
Private Sub OKButton_Click()                      '"确定"按钮的 Click 事件
    Dim str1 As String
    SearchFlag=1                                  '置查询标识为 1
    If Form1.Adodc1.Recordset.Fields(Combo1.Text).Type=202 Then
                                                  '选择字段为文本类型时
        If Combo2.Text="Like" Then                '若运算符为"Like"时
            str1="%" & Trim(Text1.Text) & "%"
        Else
            str1=Trim(Text1.Text)
        End If
        SearchString=Combo1.Text & " " & Combo2.Text & "'" & str1 & "'"
    Else
        If Combo1.Text="出生日期" Then     '若选择字段为"出生日期"时
            SearchString=Combo1.Text & " " & Combo2.Text & _
            " CDate(" & Trim(Text1.Text) & ")"
        Else
            SearchString=Combo1.Text & " " & Combo2.Text & " " & Trim(Text1.Text)
```

```
        End If
    End If
    Unload Me
End Sub
Private Sub CancelButton_Click()                    '"取消"按钮的 Click 事件
    SearchFlag = 0                                  '置查询标识为 0
    Unload Me
End Sub
```

习题十

1. 简答题

(1) 简述数据库、数据库表、记录及字段的关系。

(2) 什么是关系型数据库?

(3) 如何利用可视化数据管理器创建数据库和数据库表?

(4) 简述利用 Adodc 控件连接数据库的过程。

(5) 简述将一个 TextBox 控件绑定到 Adodc 控件的数据表中某字段的方法。

2. 填空题

(1) 数据管理技术发展过程经过人工管理、文件系统和数据库管理系统三个阶段,其中数据独立性最高的阶段是(　　)。

【出处】2005 年 9 月全国计算机等级考试二级 Visual Basic

(2) 在关系数据库中,把数据表示成二维表,每一个二维表称为(　　)。

【出处】2005 年 4 月全国计算机等级考试二级 Visual Basic

(3) 使用 SQL 语句查询数据库表"StuInfo"中所有"入学成绩"字段值大于 500 的记录应使用的语句为"Select * From StuInfo Where (　　)"。

(4) Recordset 对象的 MoveFirst 方法用于将记录指针移到(　　),MoveNext 方法用于将记录指针移到(　　),若要将记录指针移到当前记录后第 3 条记录上应使用(　　)。

(5) 若获取 Adodc 控件(名为 Adodc1)记录集中的字段数应表示为(　　),获取第一个字段的名字应表示为(　　)。

3. 选择题

(1) 数据库 DB、数据库系统 DBS、数据库管理系统 DBMS 之间的关系是(　　)。

A. DB 包含 DBS 和 DBMS　　B. DBMS 包含 DB 和 DBS

C. DBS 包含 DB 和 DBMS　　D. 没有任何关系

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(2) "商品"与"顾客"两个实体集之间的联系一般是(　　)。

A. 一对一　　B. 一对多　　C. 多对一　　D. 多对多

【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic

(3) Microsoft Access 数据库文件的扩展名为(　　)

A. doc　　B. xls　　C. mdb　　D. dbf

(4) Recordset 对象的 EOF 属性为真的情况是(　　)

A. 如果最后 1 条记录是当前记录时

B. 如果第 1 条记录是当前记录时

C. 如果最后 1 条记录是当前记录，再利用 MoveNext 方法时

D. 如果第 1 条记录是当前记录，再利用 MovePrevious 方法时

(5) 利用 Recordset 对象的(　　)方法可以添加一条新记录

A. AddNew　　B. Delete　　C. Updata　　D. MoveLast

(6) 当在程序运行期间利用以下语句来获取记录集时，应使用(　　)语句来刷新该连接。

Adodc1. RecordSource = "Select * From StuInfo"

A. Adodc1. Recordset. MoveFirst　　B. Adodc1. Refresh

C. Adodc1. Recordset. Update　　D. Adodc1. Recordset. AddNew

4. 编程题

利用 VisData 建立一个数据库及一个学生成绩表(StuScore)，编程实现一个学生成绩管理程序。该程序能够实现如下功能：

(1) 学生成绩表中包括学号、姓名、英语、高数、计算机、总成绩等字段。

(2) 程序能够实现学生成绩的录入、修改、删除、查询等功能。录入时，总成绩字段应自动根据录入的各科成绩自动获得。

(3) 查询时，能够实现按照学号、姓名、总成绩等进行查询；还能够查询某门课程不及格学生的信息以及查询所有课程中有不及格学生的信息。

第 11 章　多媒体应用程序设计

随着多媒体硬件环境和软件环境的不断完善，目前大部分计算机软件开发中都涉及多媒体软件技术的应用。

Visual Basic 提供了多个多媒体控件，如 MMControl 控件、MediaPlayer 控件等。这些控件为项目的开发带来极大的方便，使软件开发人员能灵活地控制媒体对象。

11.1　MMControl 控件

MMControl 控件管理媒体控制接口(MCI)设备上的多媒体文件的记录与回放。从概念上说，这种控件是一组按钮，它被用来向诸如声卡、MIDI 序列发生器、CD－ROM 驱动器、视频 CD 播放器和视频磁带记录器及播放器等设备发出 MCI 命令。MCI 控件还支持. avi 视频文件的回放。

MMControl 控件具有一组执行 MCI 命令的下压式按钮，如图 11－1 所示。这些按钮与通常的 CD 机或录像机上的按键类似。

图 11－1　控件的接口样式

在图 11－1 中从左向右，此按钮依次定义为到起点、到终点、播放、暂停、向后步进、向前步进、停止、录音和弹出，它们的功能是管理 MCI 设备的录制和播放。

MMControl 控件属于 ActiveX 控件而不是 Visual Basic 的标准控件，用户使用之前，需要先将其引入控件箱。具体操作如下：

(1) 选择“部件”菜单项，或按快捷键 Ctrl＋T，打开“部件”对话框。

(2) 在“部件”对话框中选中“Microsoft Multimedia Control 6.0，单击确定后，MMControl 控件出现在控件箱中。

11.1.1　MMControl 控件的属性

1. DeviceType 属性

设置多媒体控件所要打开的设备类型，其语法格式为：

MMControl1. DeviceType＝设备类型

其中“设备类型”可以是 AVIVideo、CDAudio、MMMovie、WaveAudio 等。常见的多媒体设备类型见表 11－1 所示。

表 11－1　常见的多媒体设备类型

设备类型	DeviceType 属性	文件类型	说明
CD audio	Cdaudio	音频	音频 CD 播放器
Digital Audio Tape	Dat		数字音频磁带播放器
Digital video	Digital video		窗口中的数字视频
Other	Other	未定义	未定义的 MCI 设备
Overlay	Overlay		视频重叠设备
Scanner	Scanner		图像扫描仪
Sequencer	Sequencer	.mid	音响设备数字接口序列发生器
VCR	VCR		视频磁带录放器
AVI	AVIVideo	.avi	视频文件
Videodisc	Videodisc		激光视盘播放器
Waveaudio	Waveaudio	.wav	播放数字波形文件的音频设备

2. Command 属性

在用 DeviceType 属性标识程序中要使用的设备后，就开始用 Command 属性把 MCI 命令发送给该设备。MMControl 控件使用媒体控制接口命令实现对各种多媒体设备的控制，媒体控制接口命令是一套高层次的、与设备无关的命令，其中的许多命令直接与 MMControl 控件的按钮对应。例如，Play 命令与“播放”按钮相对应。多媒体控件本质上是该命令集的 Visual Basic 接口。常用的 MCI 命令见表 11－2 所示。

表 11－2　常见的 MCI 命令

命令	MCI 命令	说　明
Open	MCI_Open	打开多媒体文件
Close	MCI_Close	关闭多媒体文件
Play	MCI_Play	播放多媒体文件
Pause	MCI_Pause	暂停播放
Back	MCI_Back	向后步进
Step	MCI_Step	向前步进
Stop	MCI_Stop	停止播放
Prev	MCI_Seek	跳到当前曲目的初始位置
Next	MCI_Seek	跳到下一个曲目的初始位置
Seek	MCI_Seek	查找曲目
Record	MCI_Record	录制
Eject	MCI_Set	弹出
Sound	MCI_Sound	播放声音
Save	MCI_Save	保存文件

3. Orientation 属性

Orientation 属性用于决定控件中的按钮是水平排列还是垂直排列。当其属性值为 1 时，各按钮垂直排列；当其属性值为 0 时，各按钮水平排列。

4. Frames 属性

Frames 属性用于指定使用 Back 命令或 Step 命令时后退和前进的帧数。

5. Filename 属性

Filename 属性指定使用 Open 命令打开或 Save 命令保存的文件名。如果在运行时改变 FileName 属性，就必须先关闭再重新打开 MCI 控件。其语法格式为：

MMControl1. FileName = 完整的文件路径及名称

11.1.2 MMControl 控件编程步骤

利用 MMControl 控件编程步骤如下：

(1) 向工具箱中添加 MMControl 控件。

(2) 利用 DeviceType 属性设置多媒体设备的类型。

(3) 若需要，则用 FileName 属性打开指定的文件。

(4) 用 Command 属性的 Open 命令打开媒体设备。

(5) 用 Command 其他命令控制媒体设备。

(6) 用 Command 属性的 Close 命令关闭媒体设备。

11.1.3 实例

【例 11-1】 创建一个简单的多媒体播放器。设计界面如图 11-2 所示。

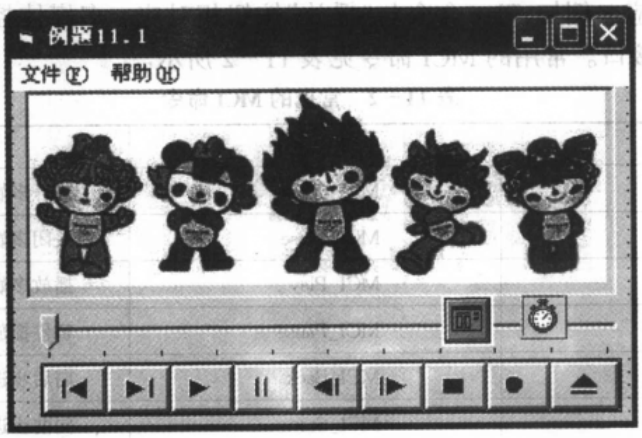


图 11-2 例 11-1 设计界面

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 建立程序用户界面。

在窗体上创建一个下拉式菜单，具体菜单名和标题设置见表 11-3。

表 11-3 例 11-1 下拉菜单属性设置

名称	标题	缩进	是否可见
mnuFile	文件(&F)		是

续表 11－3

名称	标题	缩进	是否可见
mnuOpen	打开	……	是
mnuClose	关闭	……	是
mnuExit	退出	……	是
mnuHelp	帮助(&H)		是
mnuAbout	关于	……	是

在窗体上添加 Slider 控件、MMControl 控件、CommonDialog 控件、Timer 控件和 PictureBox 控件。

(3) 进入代码编辑窗口,编写如下事件过程。

```
Private Sub Form_Load()
  Timer1.Interval=100
  MMControl1.hWndDisplay=0                    '规定显示输出的窗口为默认窗口
  CommonDialog1.DefaultExt="所有文件|*.*"     '为对话框设置默认的文件扩展名
  CommonDialog1.Filter="所有文件(*.*)|*.*|Windows 视频(*.avi)|*.avi_
  |CD 音频(*.wav)|*.wav"
End Sub
Private Sub mnuAbout_Click()
  MsgBox "多媒体播放器 V1.0"
End Sub
Private Sub mnuClose_Click()
  MMControl1.Command="Close"
  Timer1.Enabled=False
  Slider1.Value=0
End Sub
Private Sub mnuExit_Click()
  End
End Sub
Private Sub mnuOpen_Click()
  CommonDialog1.ShowOpen
  MMControl1.FileName=CommonDialog1.FileName    '多媒体控件读取打开文件
  If MMControl1.FileName="" Then
    Exit Sub
  Else
    MMControl1.Command="Open"                   '打开 MCI 文件
    Timer1.Enabled=True
    '设置 Slider 控件的最大值和最小值
    Slider1.Max=MMControl1.Length
```

```
        Slider1.Min = 0
        Slider1.LargeChange = MMControl1.Length/5
        Slider1.SmallChange = MMControl1.Length/10
    End If
End Sub
```

运行该程序，打开一个.mp3 文件，单击“Play”，即可以播放该 mp3 文件。相对应 MCI 控件中的部分就会变为有效，滑块也随之运动，如图 11－3 所示。

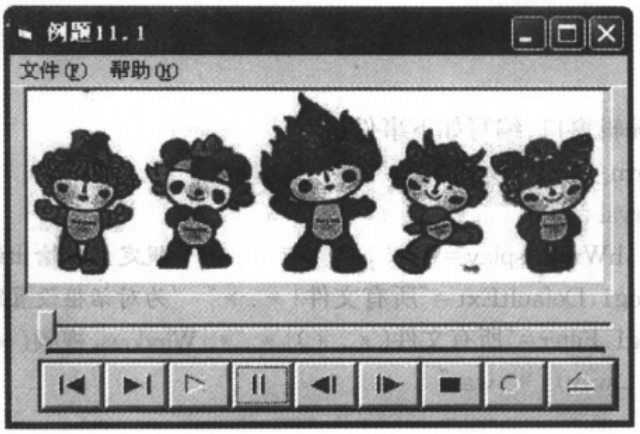


图 11－3 例 11－1 运行界面

11.2 MediaPlayer 控件

多媒体控件除了 MMControl 外，还有很多个 ActiveX 多媒体控件。例如，MediaPlayer 控件、MCIWnd 控件、Animation 控件和 PictureClip 控件。下面重点介绍广泛使用的 MediaPlayer 控件。

MediaPlayer 控件可以播放 MPEG、AVI、MIDI、WAV 和 MOV 等多种格式的多媒体文件，可以为应用程序和 Web 页添加电影和声音。与 MMControl 控件相比，MediaPlayer 控件在易用性和扩展性等方面明显占优势。它除能播放 MMControl 控件的 AVI、MIDI、WAV 文件，还能播放 MMControl 控件不能播放的 MPEG、MOV 等多媒体。MediaPlayer 控件除具有 MMControl 控件的按钮外，而且还另外自带时间数字显示和控制/显示进度滑动条。滑动条不仅能显示播放进度，而且能随意改变播放的起点。在易用性方面，MediaPlayer 控件通过执行 Play 方法非常简单地实现播放。

MediaPlayer 控件不是 Visual Basic 的标准控件，与添加其他 ActiveX 控件到工具箱的方法一样，在“部件”对话框中选择“Windows Media Player”或者通过浏览找到 Windows \ System32 目录下的 MSDXM.OCX 文件，即可添加 MediaPlayer 控件到工具箱中。

11.2.1 MediaPlayer 控件的属性

MediaPlayer 控件的重要属性见表 11－4 所示。

表 11－4　MediaPlayer 控件的属性

属　性	描　述
ShowControls	决定是否显示除“时间数字显示”之外的全部控制按钮，True 则显示
ShowPositionControls	决定是否显示位置前后移动按钮，True 则显示
ShowDisplay	决定是否显示事件数字，True 则显示
ShowTracker	决定是否显示进度控制/显示滑动条，True 则显示
AllowChangeDisplayMode	决定运行时是否允许改变显示模式
AutoRewind	决定是否自动回退
AutoStart	决定是否自动播放
Balance	设置立体声的平衡
CurrentPosition	返回当前的播放位置(单位为秒)
DisplayBackColor	指定显示面板的背景色
DisplayForeColor	指定显示面板的前景色
DisplayMode	指定显示面板的显示模式，以秒或帧显示进度。
Enabled	决定控件是否有效
EnabledContentMenu	决定是否显示快捷菜单
EnabledPositionControls	决定位置前后移动按钮是否有效
EnabledTracker	决定滑动条是否有效
FileName	指定播放所需源数据文件名
PlayCount	指定播放次数
Rate	指定播放速率
ReadyState	显示控件的准备状态
SelectionStart	指定多媒体流的开始位置
SelectionEnd	指定多媒体流的结束位置
Volumn	指定播放音量

需要特别说明的是以 Show 为首的属性之间存在制约关系，如果设置：

ShowControls＝False

ShowDisplay＝True

则仅仅显示时间数字，其余 2 个属性设置为 True 也无效(被 ShowControls 属性否决)。如果设置：

ShowControls＝False

ShowDisplay＝False

所有按钮都不显示，即 4 个属性的值均为 False。

11.2.2　MediaPlayer 控件的方法

MediaPlayer 控件的常用方法见表 11－5 所示。

表 11－5 MediaPlayer 控件的方法

属 性	描 述
Play	从指定位置或当前位置开始播放
Pause	暂停播放
Stop	暂停/停止播放，与 Pause 不同的是 Stop 方法执行语句将重置播放位置，下次播放位置取决于 AutoRewind 和 SelectionStart 属性的位置
AboutBox	显示控件的版本和版权信息
IsSoundCardEnabled	判断声卡是否有效

11.2.3 应用实例

【例 11－2】 创建一个简单的播放器。设计界面如图 11－4 所示。

图 11－4 例 11－2 设计界面

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 建立程序用户界面。在窗体上创建一个下拉式菜单，具体菜单名和标题设置见表 11－6。

表 11－6 例 11－2 下拉菜单属性设置

名称	标题	缩进	是否可见
mnuFile	文件(&F)		是
Open	打开	……	是
Play	播放	……	是
Exit	退出	……	是
Pause	暂停	……	是
Stop	停止	……	是

在窗体上添加 Windows Media Player 控件和 CommonDialog 控件。其属性值都为默认

值。

(3) 进入代码编辑窗口,编写如下事件过程:

```
Private Sub Form_Load()
    Caption = "ActiveMovie 控件示例"
    MediaPlayer1.FileName = App.Path + "\start.avi"
End Sub
Private Sub Exit_Click()
    End
End Sub
Private Sub Open_Click()
    CommonDialog1.Action = 1
    MediaPlayer1.FileName = CommonDialog1.FileName
End Sub
Private Sub Pause_Click()
    MediaPlayer1.Pause
End Sub
Private Sub Play_Click()
    MediaPlayer1.Play
End Sub
Private Sub Stop_Click()
    MediaPlayer1.Stop
End Sub
```

11.3　API 多媒体函数控制方法

在 Visual Basic 中,除使用多媒体控件实现多媒体功能以外,还可以使用 API 函数实现多媒体功能。Windows API 是操作系统支持的函数定义、参数定义和信息格式的集合,在 Visual Basic 程序中调用时首先要声明 API 函数。

11.3.1　API 函数声明

要使用一个 API 函数,首先必须在模块中声明所要使用的 API 函数。在声明 API 函数之后,可以把它当作 Visual Basic 自己的函数使用。API 函数声明格式有以下形式:

形式 1(有返回值):

Declare Function 函数名 Lib "库名" (Byval 参数 1 As 类型,…,Byval 参数 n As 类型) _
As 类型

形式 2(没有返回值):

Declare Sub 过程名 Lib "库名" (Byval 参数 1 As 类型,…,Byval 参数 n As 类型)

为了防止被声明的过程名与 Visual Basic 的过程、变量及常量发生重名或语法冲突,可在 Lib "库名"短语后面插入 Alias "别名"短语,别名指明函数或子过程在 DLL 中的真正名字。

11.3.2 API 多媒体函数

Windows 的 Winmm.dll 动态链接库为用户提供了 100 多个具有多媒体处理能力的 API 函数。例如，以 wave 开头的函数负责处理语音，以 midi 开头的函数用于完成音乐合成功能，可以用 sndPlaySound 过程播放音频文件或系统声音，还可以用与 mci 有关的高级函数来编写多媒体应用程序。下面简要介绍几个常用的 API 函数。

(1) mciExecute()

这个函数只有一个形参即 MCI 指令串，用于表明对声音文件播放的命令。

(2) mciSendString()

功能上与 mciExecute 函数相同，但它可以传送相应的信息给应用程序，使用时需要 4 个参数。第 1 个是 MCI 命令字符串，第 2 个是缓冲区，第 3 三个是缓冲区长度，第 4 个在 Visual Basic 中可恒置为 0。

(3) sndPlaySound()

此函数是一个可独立播放.wav 语音文件的函数，需要两个参数，第 1 个参数 Soundfilename 是要播放的.wav 文件的名称，第 2 个参数是一个表明播放方式的标识常量。

11.3.3 应用实例

【例 11-3】 使用 Windows API 函数 sndPlaySound 播放音频文件。运行界面如图 11-5所示。

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 建立用户界面。

在窗体上添加 2 个复选框控件、3 个命令按钮控件和 1 个通用对话框控件。各控件设置见表 11-7。

表 11-7 例 11-3 各控件属性设置

对象	属性	属性值
Command1	Name	cmdOpen
	Caption	打开文件
Command2	Name	cmdPlay
	Caption	播放文件
Command3	Name	cmdExit
	Caption	退出
CheckBox1	Name	chkSND_ASYNC
	Caption	异步播放
CheckBox2	Name	cmdAutoLoop
	Caption	循环播放
CommonDialog1	所有属性	默认

(3) 进入代码编辑窗口,编写如下事件过程:

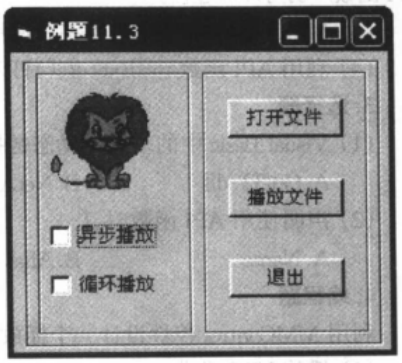


图 11-5　例 11-3 运行界面

```
'窗体通用声明
Const SND_SYNC = &H0        '同步播放
Const SND_ASYNC = &H1       '异步播放
Const SND_LOOP = &H8        '循环播放
Dim filename As String
Private Sub cmdExit_Click()
  Dim i As Integer
  filename = ""
  i = SndPlaySound(filename,1)
  End
End Sub
Private Sub cmdOpen_Click()
  CommonDialog1.DialogTitle = "选择要播放的音频文件"
  CommonDialog1.Filter = "Wave file(*.wav)|*.wav|All files(*.*)|*.*"
  CommonDialog1.Action = 1
  filename = CommonDialog1.filename
End Sub
Private Sub cmdPlay_Click()
  Dim i As Integer, flages As Integer
  Flags = 0
  If chkSND_ASYNC.Value = 1 Then
    Flags = SND_ASYNC
  End If
  If chkAutoLoop.Value = 1 Then
    Flags = Flags + SND_LOOP
    i = SndPlaySound(filename, Flags)
  End If
End Sub
```

习题十一

1. 简答题

(1) 在 Visual Basic 应用程序中怎样使用 Windows API 函数?

(2) MMControl 控件编程的步骤是什么?

(3) 在 Visual Basic 中,实现多媒体功能有几种手段?

(4) 如何声明多媒体 API 函数?

2. 填空题

(1) Windows API 是(　　)支持的函数定义、参数定义和信息格式的集合。

(2) 从 Visual Basic 应用程序中访问 Windows API 函数之前,必须在 Visual Basic 应用程序中用(　　)将

API 函数声明为外部过程。

(3) MMControl 控件中设置将要使用的多媒体设备类型的属性是(　　)。

(4) 利用 API 函数 mciExecute()打开 abc. wav 文件的语句是(　　)。

3. 选择题

(1) Visual Basic 中的多媒体动态链接库为(　　)。

A. Gdi32. dll　　B. NetApi32. dll　　C. Comdlg32. dll　　D. winmm. dll

(2) 声明使用 API 函数时,(　　)参数是需要说明的。

A. Name　　B. Alias　　C. Lib　　D. Libname

4. 编程题

利用 MMControl 控件设计一个播放器。功能包括:

(1) 播放 MP3 歌曲。

(2) 在播放 MP3 歌曲的同时同步显示歌词。

第 12 章　网络应用程序设计

Internet 编程涉及的方面非常广，平时使用的网络应用程序基本上都属于 Internet 编程的范畴。例如，使用客户端收发 E－mail，使用 FTP 客户端传输数据，使用浏览器浏览网页等，而这些基本的网络应用程序都可以用 Visual Basic 来开发完成。

12.1　网络基础知识

1. WWW

WWW 是 World Wide Web 的简称，译为万维网或全球网，是指在因特网上以超文本为基础形成的信息网。它为用户提供了一个可以轻松驾驭的图形化界面，用户通过它可以查询 Internet 上的信息资源。WWW 是通过互联网获取信息的一种方式，所浏览的网站是 WWW 的具体表现形式，但其本身不是互联网，而只是互联网的一个部分。互联网常用的服务包括：WWW、E－mail、FTP 等。

2. URL

URL 是 Uniform Resource Locator 的缩写，即统一资源定位器，也就是通常所说的网址。URL 是在 Internet 的 WWW 服务程序上用于指定信息位置的表示方法，是惟一能够识别 Internet 上具体的计算机、目录或文件位置的命名约定。

3. Port

在网络技术中，集线器、路由器的端口是指连接其他网络设备的接口。这里指的端口不是物理意义上的端口，而是特指 TCP/IP 协议中的端口，是逻辑意义上的端口。

4. HTML

HTML(HyperText Markup Language)用于创建 Web 文档的编程语言。HTML 是一种超文本标识语言，通过它可以往普通文档中加入特殊的标识符，使生成的文档中含有其他文档甚至图像、声音、动画等，从而成为超文本文档。

5. TCP/IP 协议

TCP/IP 协议及其参考模型是目前网络的标准。现在各种不同的操作平台都要依靠 TCP/IP 协议上网。其中的传输控制协议(TCP)定义了如何对传输的信息进行分组以及如何在 Internet 上传输；网际协议(IP)是一个网络编码，它既可以是网络上的一台计算机的地址，也可以是路由器一个端口的地址，即 IP 地址确定的是网络中的一个连接点，实现互联网上不同计算机之间的通信。

6. 应用层的有关协议

Internet 编程一般都需要通过某一种协议来进行，在基本 Internet 编程应用中主要使用了应用层中的 HTTP 协议和 FTP 协议。

HTTP(HyperText Transfer Protocol,超文本传输协议)是 WWW 浏览器和 WWW 服务器之间的应用层通信协议。HTTP 协议是基于 TCP/IP 之上的协议,它不仅能保证正确传输超文本文档,还能确定要传输文档中的哪一部分。

FTP(File Transfer Protocol,文件传输协议),是在两个互联网站点之间传递文件时使用的协议。FTP 是登陆到另一互联网站点的特殊途径,其登陆目的在于获取或发送文件。

12.2 Internet Transfer 控件

Internet Transfer 控件实现了两种 Internet 协议,即超文本传输协议(HyperText Transfer Protocol,HTTP)和文件传输协议(File Transfer Protocol,FTP)。使用 Internet Transfer 控件可以通过 OpenURL 或 Execute 方法连接到任何使用这两个协议的站点并检索文件。

Internet Transfer 控件的功能依赖于将要使用的协议,所以使用 Internet Transfer 控件时首先要指定要使用的协议。

为了使用 Internet Transfer 控件,需要在"部件"对话框的控件选项卡中选择"Microsoft Internet Transfer Control 6.0",确定后该控件被添加到 Visual Basic 的工具箱中。

12.2.1 Internet Transfer 控件的属性

Internet Transfer 控件的基本属性见表 12-1。

表 12-1 Internet Transfer 控件的基本属性表

编号	属性	功能
1	AccessType	设置/返回一个值,决定控件与 Internet 进行通信的访问类型
2	Document	设置/返回与 Execute 方法一起使用的文档或文件
3	hInternet	从下一级的 Wininet.dll API 返回 Internet 句柄
4	Index	设置/返回惟一的标识控件数组中一个控件的编号
5	Name	设置/返回一个控件的标识名
6	Password	设置/返回一个密码,该密码将和请求一道被发送,用以在远程计算机上登陆
7	Protocol	设置/返回一个值,指定和 Execute 方法一起使用的协议
8	Proxy	设置/返回用以和 Internet 进行通信的代理服务器的名称
9	RemotePort	设置/返回要连接的远程端口号
10	RemoteHost	设置/返回远程计算机,控件向它发送数据或从它那里接收数据
11	RequestTimeout	设置/返回在超时截止之前,按秒计算的等待时间长度
12	ResponseCode	StateChanged 事件中出现 icError 状态时,从连接返回错误码
13	ResponseInfo	返回最后发生的错误的文本
14	StillExecuting	返回一个值,指明此 Internet Transfer 控件是否处于忙状态
15	Tag	存储过程所需的附加数据
16	URL	设置/返回 Execute 或 OpenURL 方法使用的 URL
17	UserName	设置/返回与请求一起发送到远程计算机的名称

1. 访问类型 - AccessType 属性

(1) 功能:AccessType 属性设置或返回一个值,决定该控件用来与 Internet 进行通信的访问类型,访问类型可以是通过代理访问或直接访问。正在处理异步请求时,该值可以改变,但直到创建了下一个连接时,改变才会生效。

(2) 语法:

Object. AccessType = type

Object:对象,其值是 Internet Transfer 控件。

Type:整数数值,该值决定所使用的访问类型。

(3) 常用取值:AccessType 属性的取值见表 12 - 2。

表 12 - 2　AccessType 属性取值

常数	值	描　述
icUseDefault	0	默认,控件使用在注册表中找到的默认值来访问 Internet
icDirect	1	直接连到 Internet
icNamedProxy	2	命名代理,指示控件使用 Proxy 属性中指定的代理服务器

2. 文档 - Document 属性

(1) 功能:Document 属性设置或返回与 Execute 方法一起使用的文档或文件。如果未指定该属性,将返回服务器中的默认文档,如果不指定文档,则写操作会发生错误。

(2) 语法:

Object. Document = String

Object:对象,其值是 Internet Transfer 控件。

String:与 Execute 方法一起使用的文件或文档的名称。

3. 通信协议 - Protocol 属性

(1) 功能:Protocol 属性设置或返回一个值,指定和 Execute 方法一起使用的协议。指定该属性后,URL 属性被更新以显示新值。另外,如果 URL 的协议部分被更新,Protocol 属性也将被更新以体现新值。OpenURL 和 Execute 方法都可能会修改该属性值,但是,直到调用下一个 Execute 或 OpenURL 方法时,该属性值的改变才会有效。

(2) 语法:

Object. Protocol = Integer

Object:对象,其值是 Internet Transfer 控件。

Integer:整数数值,用来决定所用的协议。

(3) 常用取值:Protocol 属性的取值见表 12 - 3。

4. 远程主机 - RemoteHost、RemotePort 属性

(1) 功能:RemoteHost 属性设置或返回远程计算机,控件向它发送数据或从它那里接收数据。可提供主机名、远程计算机的 IP 地址字符串作为属性的值。RemotePort 属性则返回或设置要连接的远程端口号。在设置 Protocol 属性时,每个协议自动把 RemotePort 属性设置成适当的默认端口。

(2) 语法:

Object. RemoteHost = String

Object:对象,其值是 Internet Transfer 控件。

String:远程计算机的名称或地址。

Object. RemotePort = Port

Object:对象,其值是 Internet Transfer 控件。

Port:要连接的端口。对于 HTTP 协议该属性的默认值是 80,FTP 协议该属性的默认值是 21。

表 12-3 Protocol 属性取值

常数	值	描 述
icUnknown	0	未知的
icDefault	1	默认协议
icFTP	2	FTP,文件传输协议
icReserved	3	为将来预留
icHTTP	4	HTTP,超文本传输协议
icHTTPS	5	安全 HTTP 协议

5. 地址 - URL 属性

(1) 功能:URL 属性设置或返回 Execute 或 OpenURL 方法使用的 URL。URL 属性至少包含一个协议和一个远程主机名,且 URL 属性可以是目录或文件。

(2) 语法:

Object. URL[= URL]

Object:对象,其值是 Internet Transfer 控件。

URL:字符串,指定 Execute 或 OpenURL 方法中使用的 URL。

6. 账户信息 - UserName、Password 属性

(1) 功能:UserName 属性设置或返回与请求一起发送到远程计算机的名称。如果该属性为空,当提出请求时,该控件将把"anonymous"作为用户名来发送。Password 属性设置或返回一个密码,该密码将和请求一道被发送,用以在远程计算机上登陆。如果该属性为空,控件将发送一个默认的密码。

(2) 语法:

Object. UserName[= Name]

Object:对象,其值是 Internet Transfer 控件。

Name:字符串,指定 Execute 方法中使用的 Username。

Object. Password = String

Object:对象,其值是 Internet Transfer 控件。

String:当登陆到远程计算机时要发送的密码。

12.2.2 Internet Transfer 控件的方法

Internet Transfer 控件的方法见表 12-4。

表 12－4　Internet Transfer 控件的方法

编号	方法	功能简述
1	Cancel	取消当前请求，并关闭当前创建的所有连接
2	Execute	执行对远程服务器的请求。只能发送对特定的协议有效的请求
3	GetChunk	从 StateChanged 事件中检索数据
4	GetHeader	用于检索 HTTP 文件的标头文本
5	OpenURL	打开并返回指定 URL 的文档

1. 执行请求－Execute 的方法

(1) 功能：Execute 方法执行对远程服务器的请求。只能发送对特定的协议有效的请求。对于 HTTP 协议和 FTP 协议，Execute 方法支持不同的命令。

(2) 语法：

Object. Execute URL，Operation，Data，RequestHeaders

Object：对象，其值是 Internet Transfer 控件。

URL：可选的字符串，指定控件将要连接的 URL。如果此处未指定 URL，将使用 URL 属性中指定的 URL。

Operation：可选的字符串，指定将要执行的操作类型。其中 Execute 方法支持的 HTTP 命令见表 12－5，支持的 FTP 命令见表 12－6。

Data：可选的字符串，指定用于操作的数据。

RequestHeaders：可选的字符串，指定由远程服务器传来的附加标头。

表 12－5　HTTP 命令

运　算	描　述
GET	检索由 URL 属性指定的 URL 中的数据
HEAD	发送请求的标头
POST	传送数据给服务器，该数据在 Data 参数中。这是 GET 的替代方法，附加的指令在 Data 参数中指定
PUT	Put 操作，被替代的页面名在 Data 参数中

表 12－6　FTP 命令

运　算	描　述
CD file1	改变目录，改变到 file1 中指定的目录
CDUP	改变到父目录
CLOSE	关闭当前的 FTP 连接
DELETE file1	删除 file1 中指定的文件
DIR file1	检索 file1 中指定的目录。如果没有指定 file1，则返回当前的工作目录
GET file1 file2	检索 file1 中指定的远程文件，并创建 file2 中指定的新本地文本
LS file1	列表，搜索 file1 中指定的目录，使用 GetChunk 方法返回数据
MKDIR file1	创建目录，创建 file1 中指定的目录

续表 12－6

运 算	描 述
PUT file1 file2	复制 file1 指定的本地文件到 file2 指定的远程主机上
PWD	打印当前工作目录。访问当前目录，使用 GetChunk 方法返回数据
QUIT	终止当前用户
RECV file1 file2	检索 file1 中指定的远程文件，并创建 file2 中指定的本地新文件
RENAME file1 file2	将远程文件 file1 重命名为 fiel2
RMDIR file1	删除目录，删除 file1 中指定的远程目录
SEND file1 file2	复制 file1 指定的本地文件到 file2 指定的远程主机上
SIZE file1	返回 file1 指定的目录的大小

2. 检索数据－GetChunk 方法

(1) 功能：GetChunk 方法从 StateChanged 事件中检索数据。当 State 属性为 icResponseCompleted 时，使用 GetChunk 方法检索缓冲区的内容。

(2) 语法：

Object. GetChunk (Size[,Datatype])

Object：对象，其值是 Internet Transfer 控件。

Size：必须的。长整型数值表达式，决定被检索块的大小。

Datatype：可选的，整数，决定被检索块的数据类型，DataType 值见表 12－7。

表 12－7 DataType 值

常 数	值	描 述
icString	0	默认值，把数据作为字符串来检索
icByteArray	1	把数据作为字节数组来检索

3. 打开 URL 地址－OpenURL 的方法

(1) 功能：OpenURL 方法打开并返回 URL 的文档。文档以变体型返回。该方法完成时，URL 的各种属性将被更新，以符合当前的 URL。OpenURL 方法的返回值取决于 URL 的目标。例如，如果 URL 的目标是某个 FTP 服务器的目录，则将返回该目录；如果目标是一个文件，则检索该文件。

(2) 语法：

Object. OpenURL URL [,DataType]

Object：对象，其值是 Internet Transfer 控件。

URL：必须的，字符串表达式，指定被检索文档的 URL。

DataType：可选的，整数，用来指定数据类型，DataType 值见表 12－7。

12.2.3 Internet Transfer 控件的事件

Internet Transfer 控件的事件见表 12－8。

表 12－8 Internet Transfer 控件的事件

事 件	何时触发
StateChanged	连接中状态发生改变时发生

1.触发

在使用 Internet Transfer 控件进行网络连接的过程中，如果状态发生改变，就会引发 StateChanged 事件。一般来说，使用 StateChanged 事件决定何时使用 GetChunk 方法来检索数据。要这样做，须使用 Select Case 语句，并测试 icResponseReceived 或 icResponseCompleted。

2.事件处理过程

StateChanged 事件的处理过程为：

Object_StateChanged(ByVal State As Integer)

Object：对象，其值是 Internet Transfer 控件。

State：指定状态，取值见表 12－9。

表 12－9　State 值

常　数	值	描　述
icNone	0	无状态可报告
icHostResolvingHost	1	该控件正在查询所指定的主机的 IP 地址
icHostResolved	2	该控件已成功地找到所指定的主机 IP 地址
icConnecting	3	该控件正在与主机连接
icConnected	4	该控件已与主机连接成功
icRequesting	5	该控件正在向主机发送请求
icRequestSent	6	该控件已成功发送请求
icReceivingResponse	7	该控件正在接收主机的响应
icResponseReceived	8	该控件已成功地接收主机的响应
icDisconnecting	9	该控件正在解除与主机的连接
icDisconnected	10	该控件已成功地与主机解除了连接
icError	11	与主机通信时出现了错误
icResponseCompleted	12	该请求已完成，并且所有数据均已接收到

12.2.4　Internet Transfer 控件的应用实例

【例 12－1】 使用 Internet Transfer 控件检索 HTML，并将它显示在文本框中。程序运行结果如图 12－1 所示。

程序设计步骤如下：

(1) 新建一个“标准 EXE”工程。

(2) 选择“部件”菜单项，在弹出的“部件”对话框的控件选项卡上，选中“Microsoft Inter _ net Transfer Control 6.0”，单击“确定”按钮后 Internet Transfer 控件添加到工具箱上。

(3) 在新建窗体中添加 1 个 CommandButton 控件、1 个 RichTextBox 控件、3 个 Label 控件、1 个 Internet transfer 控件、1 个 TextBox 控件。其属性设置见表 12－10。

表 12－10　例 12－1 对象属性设置

对　象	属　性	属 性 值
Label1	Caption	地址：
Label2	Caption	状态

续表 12－10

对 象	属 性	属性值
Label3	Caption	空
CommandButton1	名称	Cmmd_query
	Caption	检索
Inet1	全部属性	默认值
RichTextBox1	Text	空
TextBox1	Text	空

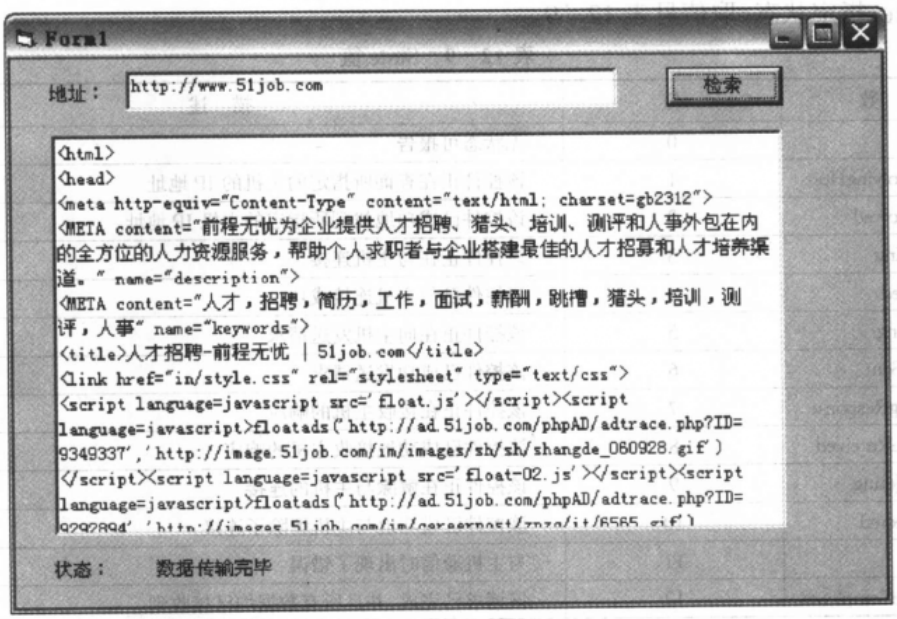


图 12－1 例 12－1 运行结果显示

(4) 进入代码编辑窗口，编写如下程序：

```
Private Sub Form_Terminate()          '程序被中断时，判断是否还在传输数据
  If Inet1.StillExecuting Then
    Inet1.Cancel
  End If
End Sub
Private Sub Cmmd_query_Click()
  RichTextBox1.Text = Inet1.OpenURL(Text1.Text, icString)
End Sub
'inet 的 statechanged 事件，根据状态不同，在 Label3 中显示不同的信息
Private Sub Inet1_StateChanged(ByVal State As Integer)
  Select Case State
    Case icConnecting
      Label3.Caption="正在建立连接…"
```

```
    Case icConnected
        Label3.Caption="建立连接成功"
    Case icReceivingResponse
        Label3.Caption="正在传输数据"
    Case icError
        Label3.Caption="传输过程中产生了错误"
    Case icDisconnected
        Label3.Caption="数据传输完毕"
  End Select
End Sub
```

(5) 程序运行,在地址栏中输入 URL,单击“检索”按钮,就会调来 HTML 文档。

12.3　WebBrowser 控件

WebBrowser 控件可以很轻松地创建 Web 浏览器,WebBrowser 控件最简单的作用就是将通过 Internet 或 Intranet 接收到的 HTML 文档转换成为所熟悉的 Web 页面。在使用 WebBrowser 控件之前,需先将该控件添加到工具箱。添加 WebBrowser 控件的方法是:选择“部件”菜单项,在弹出的“部件”对话框的控件选项卡上,选中“Microsoft Internet Controls”,单击“确定”按钮后 WebBrowser 控件添加到工具箱上。

12.3.1　WebBrowser 控件的属性

WebBrowser 控件的基本属性见表 12－11。

表 12－11　Internet Transfer 控件的基本属性表

名称	描　述
AddressBar	标识是否在浏览器上显示地址栏
Offline	标识浏览器是否处于离线浏览状态
ReadyState	标识浏览器当前处于何种状态
RegisterAsBrowser	标识浏览器控件是否被设置为一个域名解析窗口
RegisterAsDropTarget	表明是否可以把其他 ActiveX 控件或者文档放到 WebBrowser 控件上面
Resizable	标识 Internet Explorer 是否可以调节大小
Slient	标识 WebBrowser 控件是否显示对话框
TheaterMode	表明浏览器是否处于“剧场”模式
LocationName	返回 WebBrowser 控件上显示的 Web 页面的标题
LocationURL	返回 WebBrowser 控件上显示的 Web 页面的 URL 地址
Busy	该属性表明浏览器控件是否正在连接站点或下载文件,当浏览器忙的时候,用 stop 方法来强制停止当前的操作

12.3.2　WebBrowser 控件的方法

WebBrowser 控件的基本方法见表 12－12。

表 12－12 Internet Transfer 控件的基本方法表

名 称	描 述
ExecWB	执行一个 WebBrowser OLE 对象命令，并返回执行的状态
Navigate2	加载指定的 URL 或者文件路径
QueryStateWB	返回命令执行的状态
ShowBrowserBar	显示或者隐藏浏览器的工具栏
GoBack	在历史列表中向后移动一个 Web 页面
GoFoward	在历史列表中向前移动一个 Web 页面
GoHome	返回当前主页
GoSearch	转移到 Internet Explorer 中指定的搜索页面
Stop	取消当前的下载进程

1. ExecWB 方法

(1) 功能：执行一个 WebBrowser OLE 对象命令，并返回执行的状态。

(2) 语法：

Object. ExecWB nCmdID, nCmdExecOpt, [pvaIn], [pvaOut]

Object：WebBrowser 或者 Internet Explorer 对象。

nCmdID：长整型变量，要执行命令的标识符。

nCmdExecOpt：长整型变量，执行命令的选项。

pvaIn：variant 变量，命令的输入变量。

pvaOut：variant 变量，命令的输出变量。

2. Navigate2 方法

(1) 功能：用 URL 或者文件路径导航，加载所请求的网页或者文件。

(2)语法：

Object. Navigate2 URL, [Flags,][TargetFrameName,][PostData,][Headers]

Object：WebBrowser 或者 Internet Explorer 对象。

URL：一个标识要连接的地址字符串。

Flags：标志，取值见表 12－13。在使用这些标志时，可以用 And、Or 等运算符进行运算。

表 12－13 Flags 值

常 数	值	描 述
navOpenInNewWindow	1	表示在新窗口打开链接或者文件
navNoHistory	2	表示不加入历史列表
navNoReadFromCache	4	不从磁盘缓冲读取
navNoWriteToCache	8	不写入磁盘缓冲

3. ShowBrowserBar 方法

(1) 功能：显示或者隐藏 Internet Explorer 的浏览器工具栏。

(2)语法：

Object. ShowBrowserBar pvaClsid,[pvarShow]

pvaClsid:一个字符串,标识一个将要隐藏或者显示的工具栏。标准 Internet Explorer 工具栏的名称和对应的 Class ID 见表 12－14。

pvarShow:一个布尔值,设置该工具栏显示还是隐藏。

表 12－14　pvaClsid 值

浏览器工具栏名称	Class ID
Search	{30D02401－6A81－11D0－8274－00C04FD5AE38}
Favorites	{EFA24E61－B078－11D0－89E4－00C04FC9E26E}
History	{EFA24E62－B078－11D0－89E4－00C04FC9E26E}
Channels	{EFA24E63－B078－11D0－89E4－00C04FC9E26E}

12.3.3　WebBrowser 控件的事件

WebBrowser 控件的基本事件见表 12－15。

表 12－15　Internet Transfer 控件的基本事件表

名称	描　述
BeforeNavigate2	在 Navigate2 方法执行初时产生,可以用于探测是否开始浏览
DocumentComplete	当 HTML 已经显示完毕发生
DownloadComplete	在下载操作结束后触发
NavigateComplete2	在 WebBrowser 控件导航到新的 URL 后发生
NewWindow2	打开一个新的浏览器窗口时发生
OnMenuBar	当 MenuBar 发生变化时发生
OnQuit	在退出 WebBrowser 控件前发生
OnStatusBar	在 StatusBar 发生变化时发生
OnTheaterMode	当"剧场模式"属性发生变化时发生
OnToolBar	当工具栏发生变化时发生
OnVisible	当可见转向不可见或者不可见转向可见时发生
ProgressChange	当传输的进程发生变化时触发
TitleChange	在当前 Web 页面的标题发生变化时触发

12.3.4　WebBrowser 控件的应用实例

【例 12－2】 使用 WebBrowser 控件设计一个浏览器。程序运行结果如图 12－2 所示。

程序设计步骤如下:

(1) 新建一个"标准 EXE"工程。

(2) 将 WebBrowser 控件和 CommonDialog 控件添加到工具箱。

(3) 在新建窗体中添加 1 个 WebBrowser 控件、1 个 ComboBox 控件、1 个 Label 控件、1 个 CommonDialog 控件、1 个 ImageList 控件、1 个 ProgressBar 控件、1 个 StatusBar 控件、1 个 Toolbar 控件。其属性设置见表 12－16。

图 12－2　例 12－2 运行界面

表 12－16　例 12－2 对象属性设置

对　象	属　性	属 性 值
WebBrowser1	全部属性	默认
ComboBox1	Text	空
Label1	Caption	地址：
CommonDialog1	全部属性	默认
ProgressBar1	全部属性	默认
StatusBar1	全部属性	默认

此外，在 ImageList1 中加入自己喜欢的图标，并将 Toolbar1 中的命令按钮和 ImageList1 中的图标对应起来。

(4) 进入代码编辑窗口，编写如下程序：

```
Private Sub Combo1_Click()
  WebBrowser1.Navigate Combo1.Text
End Sub
Private Sub Combo1_KeyPress(KeyAscii As Integer)
  On Error Resume Next
  If KeyAscii = 13 Then
    WebBrowser1.Navigate Combo1.Text
    Combo1.AddItem CommonDialog1.FileName
  End If
End Sub
Private Sub Form_Resize()
  WebBrowser1.Width = Form1.Width - 850
  WebBrowser1.Height = Form1.Height - 2300
  Combo1.Width = Form1.Width - 2300
  ProgressBar1.Top = Form1.Height - 860
```

```
    ProgressBar1.Left = Form1.Width - 2200
End Sub
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "Back"
            On Error GoTo errback
            WebBrowser1.GoBack
            Exit Sub
    errback:
            MsgBox "前面没有 Web 页!", 64, "提示"
        Case "Next"
            On Error GoTo errfore
            WebBrowser1.GoForward
            Exit Sub
    errfore:
            MsgBox "后面没有 Web 页!", 64, "提示"
        Case "Refresh"
            WebBrowser1.Refresh
        Case "Open"
            CommonDialog1.ShowOpen
            If CommonDialog1.FileName <> "" Then
                WebBrowser1.Navigate CommonDialog1.FileName
                Combo1.Text = CommonDialog1.FileName
                Combo1.AddItem CommonDialog1.FileName
            End If
    Case "Stop"
        MousePointer = 0
        WebBrowser1.Stop
        Me.Caption = WebBrowser1.LocationName
    End Select
End Sub
Private Sub WebBrowser1_DownloadBegin()
    StatusBar1.SimpleText = "正在连接……"
End Sub
Private Sub WebBrowser1_DownloadComplete()
    On Error Resume Next
    Me.Caption = WebBrowser1.LocationName
    StatusBar1.SimpleText = "完成"
```

```
    ProgressBar1.Value=0
  End Sub
  Private Sub WebBrowser1_ProgressChange(ByVal Progress As Long, ByVal _
  ProgressMax As Long)
    If ProgressMax = 0 Then
      Exit Sub
    End If
    ProgressBar1.Max = ProgressMax
    If Progress <> -1 And Progress <= ProgressMax Then
      ProgressBar1.Value = Progress
    End If
  End Sub
```

(5) 程序运行,在地址栏中输入 URL,回车即可。

习题十二

1. 简答题

(1) 如何向工具箱中添加 Internet Transfer 控件?

(2) 如何向工具箱中添加 Web Browser 控件?

(3) Visual Basic 用于网络应用程序开发的常用控件有哪些?

2. 填空题

(1) Internet Transfer 控件实现了两种 Internet 协议,即(　　)和(　　)。使用 Internet Transfer 控件可以通过(　　)或(　　)方法连接到任何使用这两个协议的站点并检索文件。

(2) Internet Transfer 控件的常用事件包括(　　)。

(3) Internet Transfer 控件的常用方法包括(　　)、(　　)、(　　)、(　　)、(　　)。

(4) WebBrowser 控件的(　　)方法用于标识浏览器当前处于何种状态。

3. 选择题

(1) 使用 Internet Transfer 控件与标准 FTP 服务器连接时,(　　)属于文件下载的命令。

A. Copy　　B. Send　　C. PUT　　D. GET

(2) 可以使用 WebBrowser 控件的(　　)方法,来达成 Internet Explorer 浏览器的"向前"的功能。

A. GoBack　　B. GoForwad　　C. GoHome　　D. Navigate2

(3) 下面(　　)适合于实现 FTP 下载。

A. Internet Transfer　　B. WebBrowser

C. Internet Explorer　　D. ActiveX 控件

4. 编程题

自己设计题目,编程练习 Internet Transfer 控件、Web Browser 控件的应用。

附　录

附录一　Visual Basic 程序调试与错误处理

在程序设计过程中，错误是难免的。因此，查找和修改错误是必不可少的。查找和修改错误的过程称为程序的调试。Visual Basic 为程序调试提供了一组交互的、有效的调试工具和调试手段，帮助用户快捷而有效地检查出程序中错误产生的原因和地点，从而进行相应的处理。

一、常见的错误类型

Visual Basic 中的错误类型分为编译错误（语法错误）、逻辑错误、运行错误等三种。常见错误的分类、产生原因及解决方法见附录表 1。

附录表 1　常见的错误类型

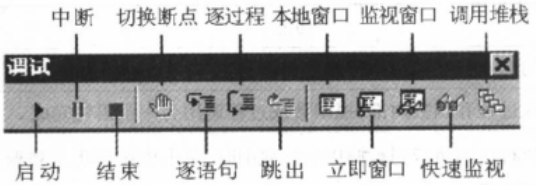
错误类型	产生原因	解决方法	说明
编译错误（语法错误）	通常是由于书写不正确引起的，例如：关键字书写错误	用 Visual Basic 中提供的自动语法检查功能	最容易发现和修改的一种错误
运行错误	运行期间代码执行了非法操作，例如：x/y 除数为 0	检查当前语句有没有出现“不合法”的情况	
逻辑错误	无具体的错误原因	一般采用“动态检查”的方法	调试程序主要就是检查逻辑错误

二、Visual Basic 的调试环境

Visual Basic 中提供了调试菜单、调试工具栏以及调试窗口等工具来帮助程序员调试程序。

1. 调试工具栏

Visual Basic 提供了一个专门用于调试的工具栏。在默认情况下，Visual Basic 界面上不显示调试工具栏。通过单击“视图”菜单→“工具栏”→“调试”，就能打开调试工具栏。工具栏中各按钮的位置如附录图 1 所示。调试工具栏各按钮的功能见附录表 2 所示。



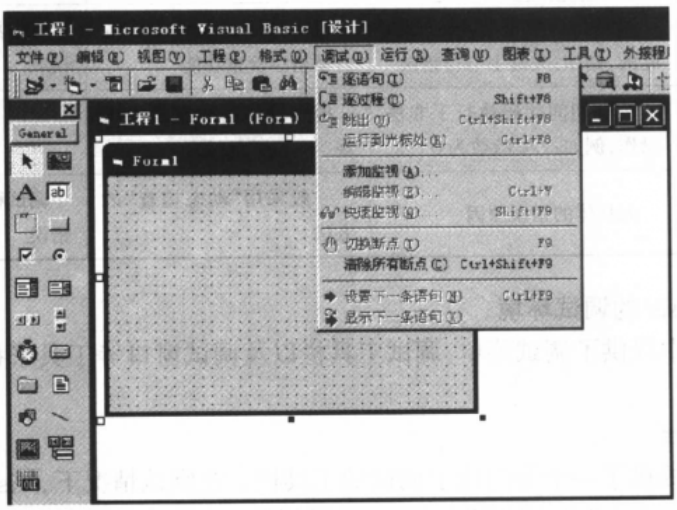
附录图 1　“调试”工具栏

附录表 2 调试工具栏中各按钮的功能

工具按钮	功能	快捷键
启动	运行程序	F5
中断	中断当前程序的运行,进入中断模式	Ctrl + Break
结束	终止当前程序的运行	
切换断点	设置或取消断点	F9
逐语句	逐行执行代码	F8
逐过程	不进入过程和函数内部,视函数或过程调用为一条语句完成	Shift + F8
跳出	逐句执行代码执行到被调用的过程中时,单击该按钮将从被调用的过程中跳出	Ctrl + Shift + F8
本地窗口	显示局部变量的当前值	
立即窗口	打开立即窗口,可以执行程序	Ctrl + G
监视窗口	打开监视窗口,显示所监视的表达式	
快速监视	将选择表达式的值显示在“快速监视”对话框内	Shift + F9
调用堆栈	可弹出一个对话框显示所有已被调用且尚未结束的过程	Ctrl + L

2. 调试菜单

Visual Basic 的调试菜单项中包含一些通用的调试命令。调试菜单的组成如附录图 2 所示,调试菜单中各菜单项的功能见附录表 3 所示。



附录图 2 “调试”菜单

附录表 3 调试菜单的功能

菜单命令	功能	快捷键
逐语句	逐语句会执行在当前执行点上的语句	F8
逐过程	不进入过程和函数内部,视函数或过程调用为一条语句完成	Shift + F8
跳出	逐句执行代码执行到被调用的过程中时,单击该按钮将从被调用的过程中跳出	Ctrl + Shift + F8

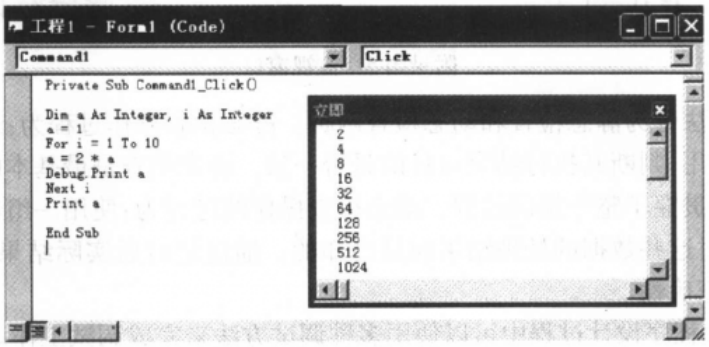
续表附录表 3

菜单命令	功能	快捷键
运行到光标处	在代码编辑窗口中，将光标连续移动到程序的某一可执行语句处，恢复中断模式	Ctrl + F8
添加监视	在设计时或是在中断模式下，显示“添加监视”对话框，输入一个监视表达式	
快速监视	将选择表达式的值显示在“快速监视”对话框内。	SHIFT + F9
切断断点	设置或删除当前行上的一个断点	F9
清除所有断点	清除所有工程中的断点	Ctrl + Shift + F9
设置下一条语句	在中断模式下，设置下一次运行时的开始语句	Ctrl + F9
显示下一条语句	突出显示下一个将被执行的语句	

3. 程序调试窗口

与程序调试相关的窗口有本地窗口、立即窗口、监视窗口三种。

(1) 立即窗口。立即窗口用于显示当前工程中的有关信息。中断模式下，用“立即窗口”调用、测试过程，在“立即窗口”会立即显示出变量或表达式的值。例如，在程序设计时，可以通过 Debug.Print 语句将中间结果显示出来，便于观察运行过程，这些中间结果就显示在立即窗口内，如附录图 3 所示。显示立即窗口的方法为单击调试工具栏上的“立即窗口”或单击“视图”菜单→“立即窗口”。



附录图 3　立即窗口

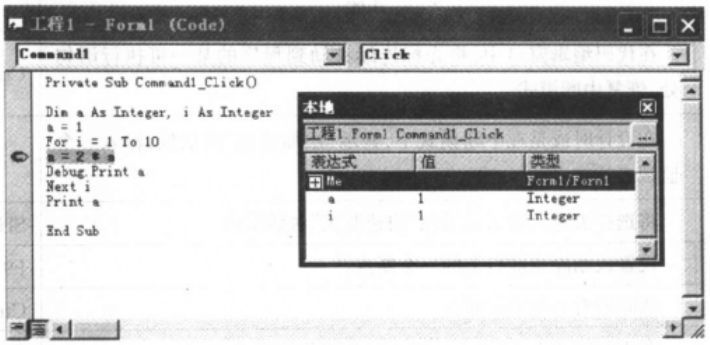
(2) 本地窗口。常与“单步执行”配合使用，本地窗口的功能是在中断模式下，显示当前过程的所有变量和活动窗体的所有属性，并能改变它们的值，如附录图 4 所示。本地窗口显示方法为：单击调试工具栏中的“本地窗口”按钮或单击“视图”菜单→“本地窗口”。

(3) 监视窗口。可以通过单击“调试”菜单→“添加监视”添加监视窗口，窗口外观如附录图 5 所示。当程序停在断点处时，为检查代码编辑窗口中变量或表达式的值，可以选中要查看的变量或表达式，然后单击“调试”菜单→“快速监视”，将打开快速监视窗口，在该窗口中能看到选中变量的值。在程序调试期间也可以添加一个监视窗口，当单步执行程序时，能够连续观察程序中各个变量的当前值，这对于调试程序是非常有用的。

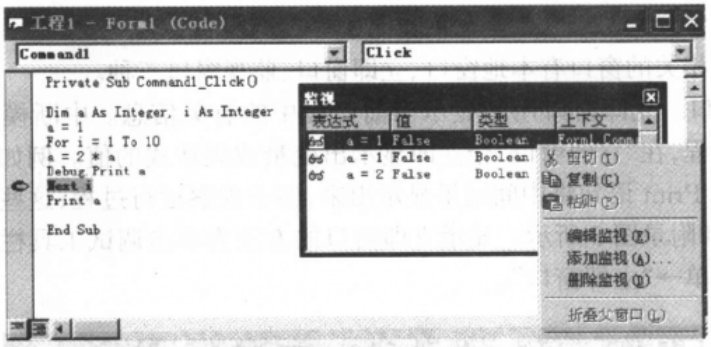
三、程序调试

在 Visual Basic 程序设计中，只有少数简单的错误(如语法错误)可以用眼睛直接看出

来,多数复杂的错误都要借助相关的调试手段来检查。



附录图 4　本地窗口



附录图 5　监视窗口

程序调试方法分为静态检查和动态检查两种。静态检查操作过程为:逐行检查语句,分析各条语句的作用,判断其执行结果与目的是否一致。静态检查是最基本的检查方法,也称为人工检查。它贯穿于整个编码过程。动态检查操作的过程为:使用一组典型的输入数据,运行程序,程序对这些数据的处理结果应是已知的。通过运行后实际结果和预想结果的比较,判断程序的正确性。

Visual Basic 程序设计过程中可以使用多种调试方法来完成调试工作。具体方法如下:

1. 设置自动语法检测

一般在安装 Visual Basic 时,系统默认设置启动自动语法检测功能。如果用户自己要进行自动语法功能设置可采用以下步骤:在 Visual Basic 集成环境中,单击“工具”菜单→“选项”命令,在打开的“选项”对话框中选择“编辑器”标签,在“代码设置”栏中选择“自动语法检测”项,单击“确定”按钮。

启动 Visual Basic 语法自动检测后,当程序员在代码窗口输入的语句有语法错误时,Visual Basic 的自动语法检测功能会立即显示错误的信息,如果忽略它,该句就会以红字显示。

2. 采用逐语句、逐过程的运行方式

逐语句运行是指一次执行一个语句,每执行一个语句就进入断点。方法为按 F8 键或选择调试工具栏的“逐语句”按钮。

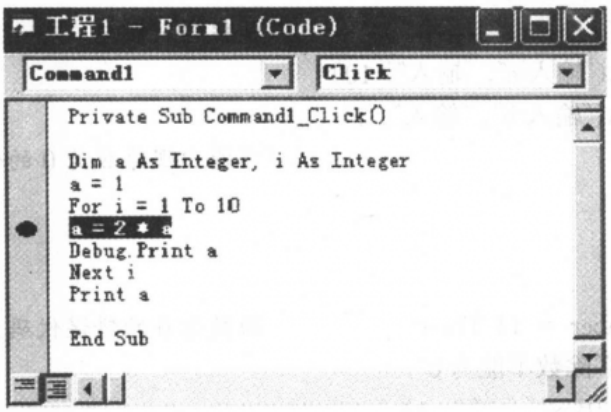
逐过程是将过程视为一个基本单位来进行调试工作。方法为按 shift + F8 或选择调试

工具栏的“逐过程”按钮。

“逐语句”命令和“逐过程”命令的区别在于后者把子过程和函数调用当作一个语句看待，不进入子过程和函数的内部，前者则进入子过程和函数的内部，逐条执行它们内部的语句。

3. 设置断点(或加入 Stop 语句)

断点是程序中作了标记的位置，如附录图 6 所示。通过断点的设置，可以使程序在需要中断的地方自动停止执行，并进入到中断模式。设置断点方法为：在代码编辑器窗口用鼠标单击要设置断点的代码行左边框位置，或单击“调试”菜单→“设置断点”，还可以在代码中加入 Stop 语句。鼠标再单击可清除断点。



附录图 6　设置断点

四、错误处理

程序的错误处理一般是先设置错误陷阱捕获错误，再进入错误处理程序，最后退出错误处理。

(1) On Error 语句用来设置错误陷阱，捕捉错误。On Error 语句的形式见附录表 4。

附录表 4　On Error 语句的用法

On Error 语句形式	含　义
On Error Goto 标号/行号	发生错误时，转到“语句标号”所指示的程序块执行错误程序，错误处理完毕，执行 Resume 语句，程序返回到出错语句处执行。若没有错误发生，过程或函数通过 Exit Sub 、Exit Function、End Sub、End Function 等语句时正常结束
On Error Resume Next	发生错误时，忽略错误行，继续执行下一个语句
On Error Goto 0	禁止当前过程中任何已启动的错误

(2) 编写错误处理程序。一般要使用到 Err 对象，它是一个系统对象，在 Visual Basic 中，可以通过 Err 对象来获取错误的消息。当出现 Visual Basic 错误时，有关错误的信息存储在 Err 对象中。Err 对象每次只维护一个错误的信息。当出现新的错误时，Err 对象将更新为新的错误信息。Error 对象属性及功能见附录表 5。

附录表 5 Error 对象的属性及功能

属性、方法	功 能
Number(错误号属性)	返回或设置运行时错误的代码
Description(错误描述属性)	返回或设置运行时错误相关的描述字符串
Source(错误来源属性)	返回或设置一个字符串,指明产生错误的对象或应用程序名称
Clear(方法)	清除对象的全部属性值
Raise(方法)	在代码中生成运行时错误,模拟产生运行时错误

【例 1】 错误捕捉程序。

```
Private Sub Form_click()
  On Error GoTo Errline                    '捕捉错误语句
  a = InputBox("输入 a", "输入", 10)
  b = InputBox("输入 b", "输入", 1)
  c = a/b                                  '可能出现除数为 0 的错误
  Print c
  Exit Sub
  Errline:
    If Err.Number = 11 Then                '除数为 0 的错误代码为 11
      MsgBox ("除数不能为 0")
      b = InputBox("", "", 1)
      Resume                               '返回
    Else
      MsgBox ("错误发生在" & Err.Source & ",错误代码为" & Err.Number & ", _
      即" & Err.Description)
    End If
End Sub
```

附录二 Visual Basic 编程规范

编程规范是指千百万有经验的程序员经历长期教训和实践后,部分程序员通过总结和反思而养成的信条和习惯。程序员使用这些规范可以提高程序的易读性、提高编程效率、获得高质量的代码。Visual Basic 编程规范涉及到程序设计中的许多方面,下面只列举部分方面的约定供参考。

一、标识符命名规范

讨论常量、变量、对象等名称的命名规范。采用如下规范利于形成良好的编程风格,使程序易于修改和阅读。

1. 变量的命名规则

变量名必须以字母开头,名字只能由字母、数字和下划线组成,名字的有效字符为 255 个;不能用 Visual Basic 的保留字作变量名,但可以把保留字嵌入变量名中,同时,变量名也

不能是末尾带有类型说明符的保留字;变量名不区分大小写。

2. 常量和变量命名约定

常量和变量需要良好格式的命名约定。下面列举了变量在命名时的约定,对于常量名应遵循与变量相同的规则。

变量的数据类型应体现在变量的名字中,对于不同类型的变量推荐使用附录表 6 所示的数据类型关键字的常用缩写。

附录表 6　数据类型关键字的常用缩写

数据类型关键字	常用缩写
Byte(字节型)	Byt
Boolean(布尔型)	Bln
Integer(整型)	Int
Long(长整型)	Lng
Single(单精度浮点型)	Sng
Double(双精度浮点型)	Dbl
Currency(货币型)	Cur
Date(日期型)	Dat
String(字符串型)	Str
Object(对象类型)	Obj
Variant(变体类型)	Var

3. 对象命名规范

在程序中应用一致的前缀来命名对象,方便用户和程序员快速、准确识别对象的类型。这部分归纳了控件及其他对象的命名规范。对象名由两部分组成,对象类型缩写和对象的定义。下面列出了 Visual Basic 支持的推荐使用的对象命名约定。

推荐使用附录表 7 所示的对象类型关键字的常用缩写。

附录表 7　对象类型关键字的常用缩写

对象类型关键字	常用缩写
CheckBox(复选柜)	Chk
ComboBox(组合框)	Cbo
CommandButton(命令按钮)	Cmd
Data(数据库控件)	Dat
DirListBox(目录列表框)	Dir
DriveListBox(驱动器列表框)	Drv
FileListBox(文件列表框)	Fil
Form(窗体)	Frm
框架(Frame)	Fra

续附录表 7

数据类型关键字	常用缩写
HScrollBar(水平滚动条)	Hsb
Image(图像)	Img
Label(标签)	Lbl
Line(线段)	Lin
ListBox(列表框)	Lst
Menu(菜单)	Mnu
OptionButton(选项按钮)	Opt
PictureBox(图片框)	Pic
Shape(形状)	Shp
TextBox(文本框)	Txt
VScrollBar(垂直滚动条)	Vsb

对于上面没有列出的控件,应该用惟一的由两个或三个字符组成的前缀使它们标准化,以保持一致性。只有当需要澄清时,才使用多于三个字符的前缀。

用户定义类型的对象,常常有必要给每种类型一个它自己的三个字符的前缀。

二、编码结构化约定

除了命名约定外,结构化编码约定可以极大地改善代码的可读性。例如,程序中加入代码注释和书写代码时采用一致性缩进等,都能提高程序的易读性。

1. 版式规范

程序书写时应采用缩进式排列,程序中用空行分隔不同的逻辑部分,用缩进分开不同的逻辑层次。

具体规则如下:

(1) 逻辑关系不紧密的部分之间要用空行分隔。

(2) 事件过程的头部、尾部与事件过程内部代码用空行分隔。

(3) 每个变量定义要尽可能占有一行,以便添加注释。

(4) 在选择结构中,语句组用向右缩进四个字符。

(5) 在循环结构中,循环体要向右缩进四个字符。

2. 代码注释规范

具体规范如下:

(1) 每一个重要变量的声明应该包括一个嵌入注释,描述该变量的使用。

(2) 变量、控件及过程的命名应该足够清楚,使得只有复杂的执行细节才需要嵌入注释。

(3) Bas 模块包含工程的 Visual Basic 一般常量声明,在其起始处,应该包括描述应用程序的综述,列举主要数据对象、过程、算法、对话、数据库及系统需求。有时,一段描述算法的伪码可能会有所帮助。

附录三　全国计算机等级考试二级 Visual Basic 考试大纲

（基础知识部分 30 分、专业语言部分 70 分）

Visual Basic 语言程序设计基本要求

1. 熟悉 Visual Basic 集成开发环境。
2. 了解 Visual Basic 中对象的概念和事件驱动程序的基本特性。
3. 了解简单的数据结构和算法。
4. 能够编写和调试简单的 Visual Basic 程序。

Visual Basic 考试内容

一、Visual Basic 程序开发环境

1. Visual Basic 的特点和版本。
2. Visual Basic 的启动与退出。
3. 主窗口：(1)标题栏和菜单。(2)工具栏。
4. 其他窗口：(1)窗体设计器和工程资源管理器。(2)属性窗口和工具箱窗口

二、对象及其操作

1. 对象：(1)Visual Basic 的对象。(2)对象属性设置。
2. 窗体：(1)窗体的结构与属性。(2)窗体事件。
3. 控件：(1)标准控件。(2)控件的命名和控件值。
4. 控件的画法和基本操作。
5. 事件驱动。

三、数据类型及其运算

1. 数据类型：(1)基本数据类型。(2)用户定义的数据类型。(3)枚举类型。
2. 常量和变量：(1)局部变量与全局变量。(2)变体类型变量。(3)缺省声明。
3. 常用内部函数。
4. 运算符与表达式：(1)算术运算符。(2)关系运算符与逻辑运算符。(3)表达式的执行顺序。

四、数据输入、输出

1. 数据输出：(1)Print 方法。(2)与 Print 方法有关的函数(Tab、Spc、Space $)。(3)格式输出(Fomat $)。
2. InputBox 函数。
3. MsgBox 函数和 MsgBox 语句。
4. 字符。
5. 打印机输出：(1)直接输出。(2)窗体输出。

五、常用标准控件

1. 文本控件：(1)标签。(2)文本框。
2. 图形控件：(1)图片框，图像框的属性，事件和方法。(2)图形文件的装入。(3)直线和形状。

3. 按钮控件。

4. 选择控件:复选框和单选按钮。

5. 选择控件:列表框和组合框。

6. 滚动条。

7. 计时器。

8. 框架。

9. 焦点与 Tab 顺序。

六、控制结构

1. 选择结构:(1)单行结构条件语句。(2)块结构条件语句。(3)IIf 函数。

2. 多分支结构。

3. For 循环控制结构。

4. 当循环控制结构。

5. Do 循环控制结构。

6. 多重循环。

7. GoTo 型控制:(1) GoTo 语句。(2)On - GoTo 语句。

七、数组

1. 数组的概念:(1)数组的定义。(2)静态数组与动态数组。

2. 数组的基本操作:(1)数组元素的输入、输出和复制。(2)For Each. . . Next 语句。(3)数组的初始化。

3. 控件数组。

八、过程

1. Sub 过程:(1)Sub 过程的建立。(2)调用 Sub 过程。(3)通用过程与事件过程。

2. Function 过程:(1)Function 过程的定义。(2)调用 Function 过程。

3. 参数传送:(1)形参与实参。(2)引用。(3)传值。(4)数组参数的传送。

4. 可选参数与可变参数。

5. 对象参数:(1)窗体参数。(2)控件参数。

九、菜单与对话框

1. 用菜单编辑器建立菜单

2. 菜单项的控制:(1)有效性控制。(2)菜单项标记。(3)键盘选择。

3. 菜单项的增减。

4. 弹出式菜单。

5. 通用对话框。

6. 文件对话框。

7. 其他对话框(颜色、字体、打印对话框)。

十、多重窗体与环境应用

1. 建立多重窗体应用程序。

2. 多重窗体程序的执行与保存。

3. Visual Basic 工程结构。(1)标准模块。(2)窗体模块。(3)Sub Main 过程。

4. 闲置循环与 DoEvents 语句。

十一、键盘与鼠标事件过程

1. KeyPress 事件。
2. KeyDown 与 KeyUp 事件。
3. 鼠标事件。
4. 鼠标光标。
5. 拖放。

十二、数据文件

1. 文件的结构和分类。
2. 文件操作语句和函数。
3. 顺序文件:(1)顺序文件的写操作。(2)顺序文件的读操作。
4. 随机文件:

(1)随机文件的打开与读写操作。
(2)随机文件中记录的增加与删除。
(3)用控件显示和修改随机文件。

5. 文件系统控件:(1)驱动器列表框和目录列表框。(2)文件列表框。
6. 文件基本操作。

考试方式

1. 笔试:90 分钟,满分 100 分,其中含公共基础知识部分的 30 分。
2. 上机操作:90 分钟,满分 100 分。

上机操作包括:(1)基本操作。(2)简单应用。(3)综合应用。

附录四　全国计算机等级考试二级 Visual Basic 样题

(【出处】2006 年 4 月全国计算机等级考试二级 Visual Basic)

一、选择题

1. 以下关于 Visual Basic 特点的叙述中,错误的是(　　)。
   A. Visual Basic 是采用事件驱动编程机制的语言
   B. Visual Basic 程序既可以编译运行,也可以解释运行
   C. 构成 Visual Basic 程序的多个过程没有固定的执行顺序
   D. Visual Basic 程序不是结构化程序,不具备结构化程序的三种基本结构
2. 以下叙述中,错误的是(　　)。
   A. 一个 Visual Basic 应用程序可以包含多个标准模块文件
   B. 一个 Visual Basic 工程可以包含多个窗体文件
   C. 标准模块文件可以属于某个指定的窗体文件
   D. 标准模块文件的扩展名是.bas
3. 以下叙述中,错误的是(　　)。
   A. 在 Visual Basic 中,对象所能响应的事件是由系统定义的
   B. 对象的任何属性既可以通过属性窗口设定,也可以通过程序语句设定

C. Visual Basic 中允许不同对象使用相同名称的方法

D. Visual Basic 中的对象具有自己的属性和方法

4. 设有如下语句：

```
Dim a,b As Integer
c="VisualBasic"
d=#7/20/2005#
```

以下关于这段代码的叙述中，错误的是(　　)。

A. a 被定义为 Integer 类型变量

B. b 被定义为 Integer 类型变量

C. c 中的数据是字符串

D. d 中的数据是日期类型

5. 以下能从字符串"VisualBasic"中直接取出字符串"Basic"的函数是(　　)。

A. Left　　B. Mid　　C. String　　D. Instr

6. 设 a=4,b=3,c=2,d=1,下列表达式的值是(　　)。

```
a>b+1 Or c<d And b Mod c
```

A. True　　B. 1　　C. -1　　D. 0

7. 以下可以作为 Visual Basic 变量名的是(　　)。

A. A#A　　B. counstA　　C. 3A　　D. AA

8. 设 x=4,y=6,则以下不能在窗体上显示出"A=10"的语句是(　　)。

A. Print A=x+y　　B. Print "A=";x+y

C. Print "A="+Str(x+y)　　D. Print "A="&x+y

9. 假定有如下的命令按钮(名称为 Command1)事件过程：

```
Private Sub Command1_Click()
  x=InputBox("输入:","输入整数")
  MsgBox "输入的数据是:",,"输入数据:"+x
End Sub
```

程序运行后，单击命令按钮，如果从键盘上输入整数 10，则以下叙述中错误的是(　　)。

A. x 的值是数值 10

B. 输入对话框的标题是"输入整数"

C. 信息框的标题是"输入数据:10"

D. 信息框中显示的是"输入的数据是:"

10. 在窗体上画 1 个命令按钮(名称为 Command1)和 1 个文本框(名称为 Text1)，然后编写如下事件过程：

```
Private Sub Command1_Click()
  x=Val(Text1.Text)
  Select Case x
    Case 1,3
      y=x*x
```

```
    Case Is<=10,Is<=-10
      y=x
    Case -10 To 10
      y=-x
    End Select
  End Sub
```

程序运行后,在文本框中输入3,然后单击命令按钮,则以下叙述中正确的是(　　)。

A. 执行 y=x*x

B. 执行 y=-x

C. 先执行 y=x*x,再执行 y=-x

D. 程序出错

11. 设有命令按钮 Command1 的单击事件过程,代码如下:

```
Private Sub Command1_Click()
  Dim a(30) As Integer
  For i=1 to 30
    a(i)=Int(Rnd*100)
  Next
  For Each arrItem In a
    If arrItem Mod 7=0 Then Print arrItem;
  If arrItem>90 Then Exit For
  Next
End Sub
```

对于该事件过程,以下叙述中错误的是(　　)。

A. a 数组中的数据是 30 个 100 以内的整数

B. 语句 For Each arrItem In a 有语法错误

C. If arrItem Mod 7=0 语句的功能是输出数组中能被 7 整除的数

D. If arrItem>90 语句的作用是当数组元素的值大于 90 时推出 For 循环

12. 设有命令按钮 Command1 的单击事件过程,代码如下:

```
Private SubCommand1_Click()
  Dim a(3,3)As Interger
  For i=1 To 3
    For j=1 To 3
      a(i,j)=i*j+i
    Next j
  Next i
  Sum=0
  For i=1To3
    Sum=Sum+a(i,4-i)
  Next i
```

```
    Print Sum
End Sub
```

运行程序单击命令按钮，输出结果是(　　)。

A. 20　　B. 7　　C. 16　　D. 17

13. 在窗体上画 1 个名称为 Command1 的命令按钮，然后编写如下事件过程：

```
Private Sub Command1_Click()
  a=0
  For i=1 To 2
    For j=1 To 4
      If j Mod 2<>0 Then
        a=a-1
      End If
      a=a+1
    Next j
  Next i
  Print a
End Sub
```

程序运行后，单击命令按钮，输出结果是(　　)。

A. 0　　B. 2　　C. 3　　D. 4

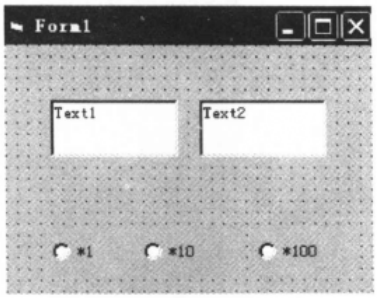
14. 窗体上有名称分别为 Text1，Text2 的 2 个文本框，有一个由 3 个单选按钮构成的控件数组 Option1，如附录图 7 所示。程序运行后，如果单击某个单选按钮，则执行 Text1 中的数值与该单选按钮所对应的运算(乘以 1、10、100)，并将结果显示在 Text2 中，如附录图 8 所示。为了实现上述功能，在程序中的问号(?)处应填入的内容是(　　)。

```
Private Sub Option1_Click(Index As Integer)
  If Text1.Text<>"" Then
    Select Case ?
      Case 0
        Text2.Text=Val(Text1.Text)
      Case1
        Text2.Text=Val(Text.Text)*10
      Case 2
        Text2.Text=Val(Text1.Text)*100
    End Select
  End If
End Sub
```

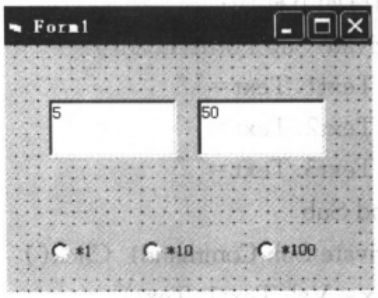
A. Index　　B. Option1.Index　　C. Option1(Index)　　D. Option1(Index).Value

15. 在窗体上画 1 个命令按钮，其名称为 Command1，然后编写如下程序：

```
Private Sub Command1_Click()
```



附录图 7



附录图 8

```
Dim a(10) As Integer
Dim x As Integer
For i=1 to 10
  a(i)=8+i
Next i
X=2
Print a(f(x)+x)
End Sub
Function f(x As Integer)
  x=x+3
  f=x
End Function
```

程序运行后，单击命令按钮，输出结果为(　　)。

A. 12　　B. 15　　C. 17　　D. 18

16. 以下关于过程的叙述中，错误的是(　　)。

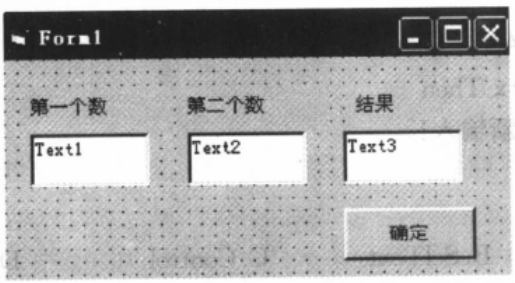
A. 事件过程是由某个事件触发而执行的过程

B. 函数过程的返回值可以有多个

C. 可以在事件过程中调用通用过程

D. 不能在事件过程中定义函数过程

17. 在窗体上画 3 个标签、3 个文本框(名称分别为 Text1，Text2，Text3)和 1 个命令按钮(名称为 Command1)，外观如图附录 9 所示。



附录图 9

编写如下程序：

```
Private Sub Form_Load()
  Text1.Text=""
  Text2.Text=""
  Text3.Text=""
End Sub
Private Sub Command1_Click()
  x=Val(Text1.Text)
  y=Val(Text2.Text)
  Text3.Text=f(x,y)
End Sub
Founction f(By Val x As Integer,By Val y As Integer)
  Do While y<>0
    Tmp=x Mod y
    x=y
    y=tmp
  Loop
  f=x
End Function
```

运行程序，在 Text1 文本框中输入 36，在 Text2 文本框中输入 24，然后单击命令按钮，则在 Text3 文本框中显示的内容是(　　)。

A. 4　　B. 6　　C. 8　　D. 12

18. 窗体上有名称分别为 Text1、Text2 的 2 个文本框，要求文本框 Text1 中输入的数据小于 500，文本框 Text2 中输入的数据小于 1000，否则重新输入。为了实现上述功能，在以下程序中问号(?)处应填入的内容是(　　)。

```
Private Sub Text1_LostFocus()
  Call CheckInput(Text1,500)
End Sub
Private Sub Text2_LostFous(  )
  Call CheckInput(Text2,1000)
End Sub
Sub CheckInput(t AS ? ,x As Integer)
  If Val(t.Text)>x Then
    MsgBox "请重新输入!"
  End If
End Sub
```

A. Text　　B. SelText　　C. Control　　D. Form

19. 在窗体上画 1 个文本框，其名称为 Text1，然后编写如下过程：

```
Private Sub Text1_KeyDown(KeyCode As Intger,Shift As Integer)
```

```
  Print Chr(KeyCode)
End Sub
Private Sub Text1_KeyUp(KeyCode As Interge, Shift As Integer)
  Print Chr(KeyCode+2)
End Sub
```

程序运行后，把焦点移到文本框中，此时如果敲击“A”键，则输出结果为(　　)。

A. A　　B. A　　C. A　　D. A
　A　　B　　C　　D

20. 为了使列表框中的项目呈多列显示，需要设置的属性为(　　)。

A. Columns　　B. Style　　C. List　　D. MultiSelect

21. 在窗体上画 1 个命令按钮，名称为 Command1，然后编写如下程序：

```
Dim Flag As Boolean
Private Sub Command1_Click()
  Dim intNum As Ingeger
  intNum=InputBox("请输入:")
  If Flag Then
    Print f(intNum)
  End If
End Sub
Function f(X As Integer) As Integer
  If X<10 Then
    Y=X
  Else
    Y=X+10
  End If
  f=Y
End Funcition
Private Sub Form_MouseUp(Buton As Integer,Shift As Integer,X As Single)
  Flag=True
End Sub
```

运行程序，首先单击窗体，然后单击命令按钮，在输入对话框中输入 5，则程序的输出结果为(　　)。

A. 0　　B. 5　　C. 15　　D. 无任何输出

22. 在菜单编辑器中建立一个名称为 Menu0 的菜单项，将其“可见”属性设置为 False，并建立其若干子菜单，然后编写如下过程：

```
Private Sub Form_MouseDown(Button As Integer,Shift As Integer,X As Single,Y _
As Single)
  If Buttion=1 Then
    PopupMenu Menu0
```

```
  End If
End Sub
```

则以下叙述中错误的是(　　)。

A. 该过程的作用是弹出一个菜单

B. 单击鼠标右键时弹出菜单

C. Menu0 是在菜单编辑器中定义的弹出菜单的名称

D. 参数 X、Y 指定鼠标当前位置的坐标

23. 窗体上有 1 个名称为 CD1 的通用对话框,1 个名称为 Command1 的命令按钮,按钮的单击事件过程如下:

```
Private Sub Command1_Click()
  CD1.FileName=""
  CD1.Filter="All Files|*.*|(*.Doc)|*.Doc|*.Txt|*.Txt"
  CD1.FilterIndex=2
  CD1.Action=1
End Sub
```

关于以上代码,错误的叙述是(　　)。

A. 执行以上事件过程,通用对话框被设置为"打开"文件对话框

B. 通用对话框的初始路径为当前路径

C. 通用对话框的默认文件类型为 *.Txt

D. 以上代码不对文件执行读写操作

24. 以下叙述中错误的是(　　)。

A. 用 Shell 函数可以执行扩展名为.exe 的应用程序

B. 若用 Static 定义通用过程,则该过程中的局部变量都默认为 Static 类型

C. Static 类型的变量可以在标准模块的声明部分定义

D. 全局变量必须在标准模块中用 Public 或 Global 声明

25. 以下关于文件的叙述中,错误的是(　　)。

A. 使用 Append 方式打开文件时,文件指针被定位于文件尾

B. 当以输入方式(Input)打开文件时,如果文件不存在,则建立一个新文件

C. 顺序文件各记录的长度可以不同

D. 随机文件打开后,既可以进行读操作,也可以进行写操作

二、填空题

1. 下列语句的输出结果是(　　)。

```
Print Format(Int(12345.6789*100+0.5)/100,"0000,0.00")
```

2. 在窗体上画 1 个命令按钮,其名称为 Command1,然后编写如下事件过程:

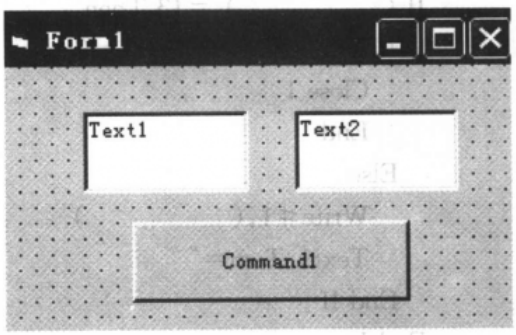
```
Private Sub Command1_Click()
  Dim arr(1 To 100) As Integer
  For i=1 To 100
    arr(i)=Int(Rnd*1000)
  Next i
```

```
Max = arr(1)
Min = arr(1)
For i = 1 To 100
  If (          ) Then
    Max = arr(i)
  End If
  If (          ) Then
    Min = arr(i)
  End If
Next i
Print "Max = "; Max, "Min = "; Min
End Sub
```

程序运行后，单击命令按钮，将产生 100 个 1000 以内的随机整数，放入数组 arr 中，后查找并输出这 100 个数中的最大值 Max 和最小值 Min，请填空。

3. 在窗体上画 1 个名称为 Command1 的命令按钮和 2 个名称分别为 Text1、Text2 的文本框，如附录图 10，然后编写如下程序：

```
Function Fun(x As Integer, By Val y As Integer)As Integer
  x = x + y
  If x < 0 Then
    Fun = x
  Else
    Fun = y
  End If
End Funtion
Private Sub Command1_Click()
  Dim a As Integer, b As Integer
  a = -10: b = 5
  Text1.Text = Fun(a, b)
  Text2.Text = Fun(a, b)
End Sub
```



附录图 10

程序运行后，单击命令按钮，Text1 和 Text2 文本框显示的内容分别是(　　)和(　　)。

4. 在窗体上画 1 个命令按钮和 1 个文本框，其名称分别为 Command1 和 Text1，然后编写如下代码：

```
Dim SaveAll As String
Private Sub Command1_Click()
  Text1.Text = Left(UCase(SaveAll), 4)
End Sub
Private Sub Text1_KeyPress(KeyAscii As Integer)
  SaveAll = SaveAll + Chr(KeyAscii)
```

```
End Sub
```

程序运行后，在文本框中输入 abcdefg，单击命令按钮，则文本框中显示的内容是(　　)。

5. 在窗体上画 1 个命令按钮和 1 个通用对话框，其名称分别为 Command1 和 CommonDialog1，然后编写如下事件过程：

```
Private Sub Command1_Click()
  CommonDialog1.(              ) ="打开文件"
  CommonDialog1.Filter="All File(*.*)|*.*"
  CommonDialog1.InitDir="C:\"
  CommonDialog1.ShowOpen
End Sub
```

该程序的功能是，程序运行后，单击命令按钮，将显示“打开”文件对话框，其标题是“打开文件”，在“文件类型”栏内显示“All Files(*.*)”，并显示 C 盘根目录下的所有文件，请填空。

6. 在窗体上画 1 个文本框，名称为 Text1，然后编写如下程序：

```
Private Sub Form_Load()
  Open "d:\temp\dat.txt" For Output As#1
  Text1.Text=""
End Sub
Private Sub Text1_KeyPress(KeyAscii As Integer)
  If (              ) =13 Then
    If UCase(Text1.Text)=(              ) Then
      Close 1
      End
    Else
      Write#1,(              )
      Text1.Text=""
    End If
  End If
End Sub
```

以上程序的功能是，在 D 盘 temp 目录下建立 1 个名为 dat.txt 的文件，在文本框中输入字符，每次按回车键(回车键的 ASCII 码是 13)都把当前文本框中的内容写入文件 dat.txt 并清除文本框中的内容；如果输入“END”，则结束程序。请填空。

三、上机操作题

1. 基本操作题

(1)在窗体 Form1 上画一个命令按钮，名称为 Cmd1，标题为“选择颜色”，在窗体上添加适当的控件并编写适当的程序代码，要求程序运行时，单击“选择颜色”命令按钮，可以弹出颜色选择对话框。程序运行界面如附录图 11 所示。

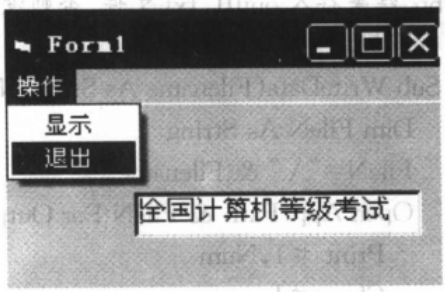
注意：保存时必须存放在考生文件夹下，窗体文件名为 jbcz1.frm，工程文件名为 jbcz1.

vbp。

(2)在窗体 Form1 上画一个名称为 Text1 的文本框,然后建立一个主菜单,标题为“操作”,名称为 Opt,该菜单有两个子菜单,其标题分别为“显示”和“退出”,名称分别为 Dis 和 End,如附录图 12 所示。编写适当的事件过程,使程序运行后,如果单击“操作”菜单中的“显示”命令,则在文本框中显示“全国计算机等级考试”;如果单击“退出”命令,则退出运行程序。

注意:存盘时必须存放在考生文件夹下,工程文件名为 jbcz2.vbp,窗体文件名为 jbcz2.frm。

附录图 11



附录图 12

2.简单应用题

在考生文件夹中有工程文件 jdyy01.vbp 及其窗体文件 jdyy01.frm,该程序是不完整的,请在有? 号的地方填入正确答案,然后删除? 及所有注释符,但不能修改其他部分。存盘时不得改变文件名和文件夹。

本题描述如下:在窗体上有 2 个单选按钮,名称分别为 Op1 和 Op2,标题分别为“宋体”和“黑体”;1 个文本框,名称为 Text1,字体为楷体_GB2312,字号为四号字;1 个命令按钮,名称为 C1,标题为“切换”。要求程序运行后,在文本框中输入“等级考试”,并选择 1 个单选按钮。在单击“切换”按钮后,会根据所选的单选按钮来切换文本框所显示的汉字字体,如附录图 13 所示。

窗体模块中的程序代码为:

```
Option Explicit
Private Sub C1_Click()
   If Op1 Then
      Text1.Font=Op1.?
   End If
   If Op2 Then
      Text1.? ="黑体"
   End If
End Sub
Pribate Sub Form_Load()
End Sub
```

3. 综合应用题

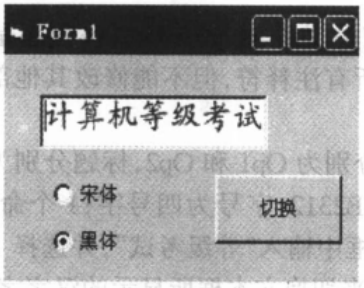
在名称为 form1 的窗体上建立 1 个文本框(名称为 Text1,Multiline 属性为 True,Scroll-

Bars 属性为 2)和 2 个命令按钮(名称分别为 Cmd1 和 Cmd2,标题分别为“读入数据”和“计算保存”)。要求程序运行后,如果单击“读入数据”按钮,则读入“in01.txt”文件中的 100 个整数,放入一个数组中(数组下界为 1),并在文本框 Text1 中显示出来;如果单击“计算保存”按钮,则把数组中的前 50 个数据在文本框 Text1 中显示出来,并存入考生文件夹中的文件“out01.txt”中,(在考生文件夹中有标准模块 mode01.bas,考生可以把该模块文件添加到自己的工程中)。程序运行界面如附录图 14 所示。

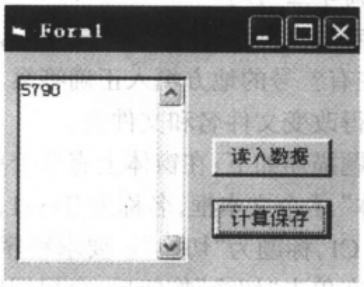
注意:文件必须存放在考生文件夹下,窗体文件名为 zhyy01.frm,工程文件名为 zhyy-o1.vbp,结果存入 out01.txt 文件,否则没有成绩。

标准模块中的程序代码为:

```
Sub WriteData(Filename As Sring,Num As Long)
  Dim FileN As String
  FileN="\" & Filename
  Open App.Path & FileN For Output As #1
    Print #1,Num
    Close #1
End Sub
```



附录图 13



附录图 14

参 考 文 献

1. 白康生. Visual Basic 程序设计. 北京:清华大学出版社,2006
2. 李书琴. Visual Basic 程序设计基础. 北京:清华大学出版社,2006
3. 王萍. Visual Basic 程序设计基础教程. 北京:清华大学出版社,2006
4. 明日科技. Visual Basic 程序开发范例宝典. 北京:人民邮电出版社,2006
5. 汤涌涛. Microsoft Visual Basic 2005 从入门到精通. 北京:清华大学出版社,2006
6. 章立民研究室. Visual Basic 2005 程序开发与界面设计秘诀. 北京:机械工业出版社,2006
7. 郑阿奇. Visual Basic 教程. 北京:清华大学出版社,2006
8. 张增良. Visual Basic 程序设计简明教程. 西安:西安交通大学出版社,2006
9. 匡松. Visual Basic 程序设计基础教程. 北京:中国铁道出版社,2006
10. 夏邦贵. Visual Basic 6.0 数据库开发经典实例精解. 北京:机械工业出版社,2006
11. 李春葆. Visual Basic 程序设计. 北京:清华大学出版社,2005
12. 麻新旗. Visual Basic 大学基础教程. 北京:电子工业出版社,2005
13. 王晟. Visual Basic. 数据库开发经典案例解析. 北京:清华大学出版社,2005
14. 孙俏. Visual Basic 程序设计. 北京:中国铁道出版社,2005
15. 王温君. Visual Basic 程序设计教程. 北京:清华大学出版社,2005
16. 陈学东. Visual Basic 6.0 程序设计教程. 北京:清华大学出版社,2005
17. 梁普选. 新编 Visual Basic 程序设计教程. 北京:电子工业出版社,2005
18. 赛奎春. Visual Basic 工程应用与项目实践. 北京:机械工业出版社,2005
19. 曾强聪. Visual Basic 程序设计基础教程. 北京:清华大学出版社,2004
20. 王兴晶. Visual Basic 6.0 应用编程 150 例. 北京:电子工业出版社,2004
21. 李华飚. Visual Basic 数据库编程从范例入门到项目开发. 北京:人民邮电出版社,2004
22. 刘新民. Visual Basic 6.0 程序设计. 北京:清华大学出版社,2004
23. 罗朝胜. Visual Basic 6.0 程序设计教程. 北京:人民邮电出版社,2004
24. 高春艳. Visual Basic 数据库开发关键技术与实例应用. 北京:人民邮电出版社,2004
25. 宁正元. Visual Basic 程序设计教程. 北京:北京交通大学出版社,2004
26. 林卓然. Visual Basic 程序设计教程. 北京:电子工业出版社,2004
27. 张勇. Visual Basic 课程设计案例精编. 北京,中国水利水电出版社,2004
28. 刘新民. Visual Basic 6.0 程序设计. 北京:清华大学出版社,2004
29. 张基温. Visual Basic 程序开发教程. 北京:清华大学出版社,2004
30. 甄彤,陈卫东. Visual Basic 程序设计及教程. 北京:机械工业出版社,2004